SEGA® COMPUTER

The Official Sega User Club Magazine

NOV 1986 / FEB 1987

In this Issue:
TECHNICAL DATA ABOUT LSV
LOADING ZX SPECTRUM PICTURES
AIRCRAFT
AMAZE
AUSTRALIAN CONTRIBUTORS CORNER
NEWS & REVIEWS
SEGA PROGRAMMING — PARTS 3, 4 & 5

SEGA®
COMPUTER

The official magazine of the Sega User Club, New Zealand.

Registered as a magazine at the CPO Wellington, New Zealand. Published bi-monthly. Magazine published about the middle of July, Sept, Nov, Jan, Mar, May.

Published by:
Sega Software Support
P.O. Box 10-207
Auckland, 4.
NEW ZEALAND

Telephone (09) 603-274

Editor: Glenys Millington

All contributions welcome. Name, address and phone number must be included.

All software programs published by the magazine become the property of Sega Software Support unless by prior arrangement. They are accepted on the basis that they are the original work of the author. The programs must be submitted on tape or disc and a printed listing is desirable. Software is not returned unless accompanied by a stamped self addressed envelope.

For overseas contributions, please enclose a $1 note/coin of the country of origin if software is to be returned.

A plea for software listings - please check your software thoroughly for errors and spelling before sending it to us. Please update us on any errors you know about so that we can publish corrections.

We pay $NZ20.00 for the feature software program published each month and $NZ4.00 for all other full programs published.

Articles or reviews must be legible. Articles or reviews are paid $NZ4.00 per page on a prorata basis.

All contributions are subject to approval by the editor and may be edited to suit the magazine style. All payments are made after publication.

SEGA USER CLUB
MEMBERSHIP YEAR JULY 1986 - JUNE '87

Membership includes a subscription to the Club magazine SEGA COMPUTER as well as qualifying for other club benefits.

NEW ZEALAND
Membership Subscription: $40.70 incl. GST.

Back copies available:
$7.15 each, $35.00 complete 1985/86 year

AUSTRALIA
Membership Subscription:
$NZ50.00 Airmail, $NZ46.00 Surface Mail

ELSEWHERE
Membership Subscriptions available on request.

CONTENTS

Letters to the Editor 2
News & Reviews 4
Caverns of Karanor (review) 5
Technical Data About LSV 6
Loading ZX Spectrum Pictures . . . 8
The SEGA Computer
Part 3 Sound Generator 10
Part 4 Cassette Routines 11
Part 5 Keyboard & Joysticks . . 13
Aircraft 15
Amaze 22
Hackers Corner 25
Mr Moo 26
Statistical Programs
by Dr D. A. Campbell 27
Programs by B. E. Pycroft 30
Big Characters 31
Australian Contributors Corner . . . 32

EDITORIAL

Christmas has once again been and gone and we are now well into 1987. With all the difficulty of trying to get a magazine out in the Christmas rush we decided to put out a double issue to start off the new year and to bring the magazine right back into it's right time-frame.

A marvellous Christmas was had by all at our place, with 14 members of the family staying. Christmas morning saw an absolute mountain of presents under the tree and an even bigger mountain of food to be demolished.

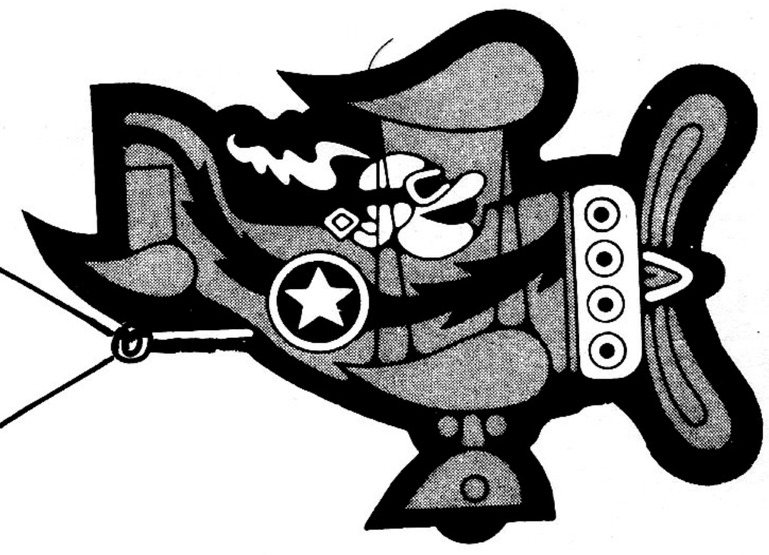
Our 10 days away were spent sailing around the Marlborough Sounds, away from it all. Absolutely glorious!

This edition sees a variety of programs, including contributions from some of our Australian members, the continuation of "The Sega Computer", a review of one of our games by one of our younger members, and some brief comments on the results of the survey sheets you all completed.

From all the staff at Sega Software — Happy 1987! — may it be happy and prosperous for you all.

Glenys Millington
Editor

LETTERS ... to THE EDITOR.



DEAR EDITOR
Like others I am frustrated by not being able to run "Jet Ranger" on a 32K cartridge. It seems to me that the answer must lie somewhere in the POKE or CALL addresses for the machine code. Can anybody help — or do I just scrub the hard work of typing it all in?
B.C.R. Davis, Bucklands Beach.

DEAR EDITOR
I noticed in the July/August magazine that the Basic keyword MOTOR appeared in Table XXX2. What is the keyword used for and in what circumstances?
N. Hanson, Dunedin.

EDITOR'S REPLY
MOTOR is used to control the cassette port. It does not seem to be implemented in Sega Basic.

DEAR EDITOR
I am a member of the club and have found the mags of late very good, but can you please tell me how to use commands: *Save v "****", start, end, *Save c "****" start, end and what it means by: IF CHR$(0). Please tell me what I have to do with these.
Denver Scott, Bucklands Beach.

EDITOR'S REPLY
Contact Mike Hadrup on Auckland 534-3379.

DEAR EDITOR
I am writing to you with a few small questions and ideas for your excellent magazine.

1. Where can I buy a copy of Brian Brown's Sega machine code book? I know you are publishing a series of articles from it, but I would like a copy.

2. Why don't you have in your magazine a "hints and tips" page for games and things for the Sega? You could ask people to send in maps, hints, ideas, etc., for the Sega games software and you could publish them in each issue. (I myself could send you maps and hints on most of the Sega adventures, lots of the cartridges and cassette software.)

If this idea was carried out it would help many people get further on and enjoy their games more.
David Downs, Wanganui.

EDITOR'S REPLY
I do not know of any outlet that still has Brian Brown's book in stock. With regard to your suggestion on hints and tips for games — a good idea — maps however, would not be suitable for reproduction by the printers.

DEAR EDITOR
Could anyone supply me with information, software and circuits for modems or interface to use the Sega as an RTTY terminal.

I know various other computers have this software, etc., available, but as I have had my Sega since July 1984, I would like to use it in this mode.
J. Lindsay, Pakuranga.

EDITOR'S REPLY
Can anyone help?

DEAR EDITOR
Three small printing errors in the random-shuffle routine in the last magazine. Correct lines are:

```
80 IR$ = "": IF LEN(I$) > J THEN IR$
   =RIGHT$(I$,LEN(I$) — J)
120 FOR K = 1 TO LI
   130 M$ = M$ + J$(K)
```

Chris Rodliffe, Auckland.

EDITOR'S REPLY
Our apologies!

DEAR EDITOR
1. The Sega seems to be coming of age, thanks to the routines of Michael Hadrup. I have read plenty of extension programs for other computers, V12, C64, ZX Spectrum, but not knowing enough about machine code or the Sega ROM routines I could not see how to extend Sega Basic.
(a) Would Michael Hadrup be willing to write an explanatory article detailing his routines? Does he revector the error messages to include his new routines?
(b) Perhaps Michael could look at writing file handling routines for cartridge Basic, and tape recorders, along the lines of Print #1, Input #1, etc.

2. I don't know if this will help Chris Rodliffe, but my first Euromatic was returned for repairs due to erratic operation, the volume level had to be set lower and lower, until it no longer worked. FTC actually replaced the dataset with another Euromatic and this one is generally reliable at volume setting 6-7 but it occasionally requires lower settings to load programs.

I look forward to receiving more great magazines.
D. Coursey, Christchurch.

EDITOR'S REPLY
(a) Yes — more in later issues.
(b) Unlikely — ring Michael Hadrup, phone number listed above.

DEAR EDITOR

In the July/August magazine you answered my letter in which I wanted a program for producing anagrams.

While in Australia on holiday I gave the problem to my 16-year-old nephew and he came up with a program in about a quarter of an hour on his Apple II which with one small adjustment works on the Sega and is printed below:

```
  5 Rem Anagrams (Input word is EARTH)
 10 FOR T = 1 TO 5
 20 R = INT(RND(T)*5)+1
 30 ON R GOTO 40,50,60,70,80
 40 IF W1<1 THEN PRINT "E";:W1=1 : GOTO 100
 45 GOTO 20
 50 IF W2<1 THEN PRINT "A";:W2=1 : GOTO 100
 55 GOTO 20
 60 IF W3<1 THEN PRINT "R";:W3=1 : GOTO 100
 65 GOTO 20
 70 IF W4<1 THEN PRINT "T";:W4=1 : GOTO 100
 75 GOTO 20
 80 IF W5<1 THEN PRINT "H";:W5=1 : GOTO 100
 85 GOTO 20
100 NEXT T
105 W1=0 : W2=0 : W3 = 0 : W4 = 0 : W5 = 0
110 PRINT: GOTO 10
```

If you have a word with two letters the same, e.g. instead of "earth" you input "three" the appropriate line would be: IF W4<1 THEN PRINT "E";: W4=W4+.5 : GOTO 100 and delete W5 or you will get more than two E's.

P.S. I was delighted to receive the SP400 Printer Plotter for the "Sega is Supreme" competition.

B.A. Turnell, Waikeria.

DEAR EDITOR

With reference to the reported problems with "3-D Noughts and Crosses" these are caused by one of the "classic" boo-boo's of programming — that of jumping out of a FOR....NEXT loop without properly terminating it.

When the variable concerned is used again as the INNER variable of a Nested Loop then the program crashes with a "Next Without For" error on the OUTER variable. (Since the error appears on the "wrong" variable, it can be very tricky to find the culprit.)

In the reported case, the error occurs in the loop in lines 1590-1690, with the jump-outs in lines 1660, 1670, and 1680.

The relevant lines from the program are:

```
1590  FOR K = TO 18
1680  IF P<10 THEN 1720
1690  NEXT K
1720  (REM)
2310  FOR J=1 TO 4
2350  FOR K=1 TO 4
2450  NEXT K
2470  NEXT J
```

Running this program gives the reported error "Next Without For In 2470" (i.e. the error appears to be on J).

This error can be eliminated by replacing the FOR... NEXT loop in lines 1590-1690 with an IF... GOTO... The two lines to change are:

```
1590  K = 1
1690  K=K+1:IF K <19 THEN 1600
```

There is a similar error in lines 2160-2180 which will produce a "Next Without For in 2480". To cure this one, change these lines:

```
2160  J=A
2180  J=J+5-2*A:IF J <10-3*a THEN 2170
```

The exact nature of the consequences of jumping out of a loop are rarely spelt out in programming texts and the cause of the fault is not obvious.

G.R. Petherick, Auckland.

DEAR EDITOR

As usual, I was the 5th person to read *Sega Computer* for October in our household!

1. Hucal. The undermentioned secret to sorting is that the cursor should be placed at the top left corner of the block of information to be sorted. I use Hucal for my genealogical research and stamp collection.

2. K. Nightingale's printer problem. This is a problem that all "Sega Basic" systems have! Enclosed is a copy of "Comset" routine correction to permit the expected operation of the RS232 port.

3. Character patterns. Enclosed is a list of patterns starting at &H1A08 in VRAM. These may be copied onto a initialising program or simply direct entry if only a few are involved. Good programming should not require the Primary character set to be altered as there is an alternate 126 to be utilised.

4. C. Rodliffe's Euromatic. The Sega is sensitive to wave shape and stability, so recorder head azimuth might need adjusting. Also a cotton bud and meths or alcohol make good head cleaners, an important requirement for any recorder. I have had a successful load from an ordinary player, a program that would not load from my Sega dataset!

5. If D. Coursey were to peruse a book on the Z80 microprocessor, he/she might get a better feel for machine code. The 'CB' and 'ED' are prefixes to the hex code to give the mnemonic indicated in that column.

Enclosed are a number of programs. The directory presentation should be a good asset to Basword.

I enjoy *our* magazine, especially as there appears to be increasing input on the intricacies of Sega and machine code.

It appears that M. Howard has left publishing for Sega. What a pity — he started something useful.

P.S. Would it be pertinent to publish the addresses of those who have their contributions published? Please do with mine. A useful information interchange could result by direct communication.

B.E. Pycroft, 261 Wellington St,
Wainuiomata. Ph (04) 646-790.

EDITOR'S REPLY

1. Thank you for the useful and informative comments. Addresses of contributors can only be published with their consent.

2. Michael Howard was at varsity last year and therefore not always available to do a column. However, we do have another contribution from Mike in this issue.

3. Most of the programs mentioned in your letter are published further on in the mag.

4. Your earlier comments regarding the 255 colour hues appears to be correct. Unfortunately we no longer have access to the original content of the magazine (when Grandstand were publishing it) to rectify the error.

5. Regarding your confidential comments — a great idea — do you know of someone that can do it.

DEAR EDITOR

Could you please supply me with some info on the game "Lode Runner"?

Michael Wilkinson, Opotiki.

EDITOR'S REPLY

"Lode Runner" has not been available in New Zealand for quite some time now and is unlikely to be available again.

The special offers to members in the last magazine brought in an enormous pile of mail in the two weeks leading up to the Christmas break, and all the staff worked right through to the 19th December to ensure that everything was processed in time. We had hoped to close on December 16, but those Christmas stockings needed to be filled. I hope yours was.

Just prior to Christmas one of our representatives went to Japan in an attempt to achieve a better pricing system for new software for the Sega. This was not to be! Prices for ordinary cartridges have doubled, so in the meantime we will continue to market the stocks we have on hand and locally written software.

The "Sega Is Supreme" survey sheets were returned to us in their droves. Although comments were many and varied, only three major areas of dissatisfaction were apparent:

1. No retail outlets for Sega software
This is unfortunately out of our hands. Although we are now the only major distributor of software for the Sega in Australasia we still buy direct from a wholesaler and therefore our hands are tied by contracts.

2. No new software
Refer to our comments above and also those in the Sept/Oct issue regarding the prices of importing new software from Japan. I would like to make mention here of some of the excellent locally written software which is in some cases superior to the software in Japan.

3. Magazine delays
This problem dates back a long time and all our efforts to overcome this problem seem to have been in vain. This should now be rectified with this double-issue bringing the magazine back into the right time-frame.

It was heartening to read favourable comments on the improvement in service, and in the magazine, since we took over last July. Interesting to note that our Australian members have bigger problems obtaining software over there than we New Zealander's do.

In this issue we have a review of a locally written game from one of our members in Hawera. Should anyone else be interested in doing something similar, please do not hesitate. After all, it's your contributions that keep the magazine going.

A REVIEW OF
CAVERNS OF KARANOR

By Karl Woodhead

Well it looks like Michael Boyd has done it again. Not content with the success of "Burglar Bill", he has gone and helped the Sega software scene even more.

This time Bill's mission is to save the universe by capping 11 tanks in the caverns below the ancient Temples of Karanor. Deadly mutants are escaping from these tanks and threaten life as we know it. None have escaped from the caverns as yet, but if Bill doesn't act quickly they will.

You have only three lives to complete the game with. You also have a limited amount of energy which goes steadily down as you progress through the game. There are places you can go to restore your energy back to full, just where they are we'll let you find out.

This is one of the best games I have played on the Sega. What I like about this game is you can walk back and forth through most of the rooms with ease. To go into some rooms you need certain objects to move past the exits.

The graphics in this game are amazing, with smooth animation. It is 100% machine code. "Caverns of Karanor" has got some fiendlishly hard puzzles which makes a must for your collection.

Ratings:	Graphics	10	Value	9
	Sound	7	Playability	9

How the ratings are graded:

GRAPHICS: Is the game graphically great — or do the graphics grate? This mark puts you in the picture and lets you know if things are sprite and beautiful.

SOUND: Is the sound loud and proud, or does it sound like you've got your head in a bucket of mud?

VALUE: Should you beg, borrow — but never steal — to get the cash for this game, or should you wait until it turns up in the bargain basement after a week?

PLAYABILITY: Will you have to be dragged screaming from the joystick after 24 hours — and still be anxious for more, or will you finish it in a few minutes?

Technical Data about LSV

By Michael Hadrup

A number of ROM routines and system variables (RAM routines) are used by LSV to extend the Basic interpreter. These include evaluation, execution and some multi purpose routines. They are listed below with Disk Basic and Cartridge addresses, and a short description of what they do.

ROM Routines

Disk Basic	Description	Cartridge Basic
&H4B17	This routine prints a message stored at an address in the HL register pair ending in a null CHR$(0), on the current display.	&H4E52
&H7BB6	This routine prints a character in the A register on the current display.	&H2B52
&H1AA5	This routine prints an error message defined by a number in the A register on the current display.	&H73B7
&H3BD	This is an entry point in the Basic start up that prints bytes free, resets the system and enters the Editor.	&H6A55
&H7C2	BREAK routine - This routine is used when break is pressed during program execution or listing etc. It stores the state of the system (for a CONT) and resets the display to SCREEN 1,1, prints BREAK or BREAK IN line no and returns to the Editor.	&H6DE6
&H2192	Jump BREAK - This routine is used to jump to the BREAK routine using the BREAK vector.	-
&H748	ERROR routine - This routine is used when an error occurs during program execution. It resets the display to SCREEN 1,1, prints the error and returns to the Editor.	&H6D81
&H7D53	Jump ERROR - This routine is used to jump to the ERROR routine using the ERROR vector.	-
&H623C	NEW routine - This is part of the NEW command. It resets all the Basic pointers.	&H3D26
&H5E0	CONT routine - This routine is part of the CONT command. It resets the system to the state it was in after the last break, (if it existed) and continues program execution.	&H6C43
&H5D3	RUN program - This routine is part of the RUN command. It clears variables and stacks then RUN's the program.	&H6C37
&H58F	RUN program - This differs from the previous routine as it RUN's the program at the line no. in the HL register.	&H6BF1
&H4659	GOTO line - This routine does not clear the variables or stacks, but GOTO's the line no in the HL register.	&H6788
&H716	This routine is the return point during program execution, after the completion of a Basic command routine. It determines if there are more commands to execute.	&H6D4F
&H843E	This routine skips spaces using DE as a pointer, (in a Basic program).	&H2310
&H843D	This routine moves DE past one character then skips spaces. (in a Basic program).	&H230F
&H83F2	This routine evaluates a line no using DE as a pointer (in a Basic program), into the HL register	&H22C4
&H4AFE	This routine evaluates a memory address using DE as a pointer (in a Basic program), into the HL register.	&H4E39
&H6838	This routine evaluates a string expression using DE as a pointer (in a Basic program), into the character string operation stack.	&H2C76

&H74EB	This routine copies a string from the character string operations stack to a string register defined by the A register (A must hold 1 or 2).	&H2489
&H83E1	This is a routine for checking a range of values in the HL register against the A register. The carry flag is reset if H >= A <= L.	&H22B3
&H7061	This routine jumps to the (A-1) address in a table following the CALL. This is used throughout Basic to execute command routines.	&H339B
&H152C	This routine is used to pad a filename with spaces which is stored in string register 1. It pads the name or truncates it to 16 characters.	&H7B13
&H566A	This is the key scanning routine. Any key pressed except BREAK and RESET, is returned as its ASCII value. A null CHR$(0) denotes no key pressed.	&H436D
&H51B	This is the BOOT routine in Disk Basic or reset for Cartridge Basic.	&H0
&H383E	This routine is only used in Disk Basic. It turns off the disk drive.	-

System Variables or RAM Routines

&HAD5B	BREAK vector - This can be changed to vector the BREAK key.	&H95A4
&H9ED5	ERROR vector - This can be changed to vector any ERROR's.	&H86E8
&HAB16	This holds the current display mode.	&H9336
&HAB19	This holds the text screen colour.	&H9339
&HABE1	This holds the graphics screen back drop colour.	&H9410
&H9A8F	This holds the length of string register 1.	&H82A2
&H9A90	This is the start of string register 1.	&H82A3
&HAECD	This holds the character under the cursor (during the flash). When poked to a null this supresses the extra character printed when break is pressed.	&H9714
&H9954	Start of Basic program.	&H8160
&H9956	Start of Basic array storage area.	&H8162
&H9958	Start of Basic variables storage area.	&H8164
&H995A	End of Basic storage area + 1.	&H8166
&H995C	Limit for Basic storage area + 1.	&H8168

The following errors are produced by LSV...

Out of memory error -	If there is not enough memory available when a program is loaded.
Tape read error (Disk Basic) -	When a loading error occurs.
Memory writing error (Cartridge) -	When a loading error occurs ; This was used because Tape read error does not exist as an error but as a message produced by the loading routines.
Command parameter error -	If the parameters following an extended command do not comply with the correct syntax

If an error occurs, the error number is stored at &HFB66. If errors are vectored, a Basic program can determine what error occured by using PEEK(&HFB66) If you wish to print the error, a small routine has been incorporated at &HFB9B for this purpose. The error is printed on the current display by CALL &HFB9B.

If you wish to disable the "Start tape and press any key" message then use POKE &HFB3C,0.

Clever use of the SCREEN command will supress the printing of the filename during loading. An alternative on the graphics screen is to use CURSOR 0,185.

More information will be provided in the next magazine, on how to add your own extended BASIC commands. The information provided should be enough for some of you budding machine code programmers to understand how LSV works.

Loading ZX Spectrum Pictures

(C) 1986 MJH SOFTWARE

This program requires the Load/Save/Verify Basic extension in the previous magazine. If you haven't typed them in yet, then get to it! If you have typed them in, then Load and initialise them.

This program will load and convert any normally saved SPECTRUM picture. If you have a friend with a SPECTRUM then borrow some of his games tapes.
Type in the listing and save it using

*SAVE "ZX Convert.Dta" or SAVE "Convert.Dta"

RUN the program. If you wish to save the code then type

*SAVEC "ZX Convert.Cde",&HF800,&HF882 or SAVEM "Convert.Cde",&hF800,&HF882

Locate a SPECTRUM picture, and position the tape before the main block. (Just after the header, if the picture has one). Now type

CALL &HF870 and press PLAY on the TAPE

You should see and hear the loading of the picture. Now type

SCREEN2,2:CALL &HF810 and press Shift-Break

The picture you loaded should now have appeared on the graphics screen. This picture can now be saved using

*SAVES "Filename" to save the SEGA version or

SAVEM "Filename",&HC000,&HDAFF to save the SPECTRUM version.

With the use of Printer Shades in the previous magazine it is possible to dump these pictures to a printer.

Listing 1

```
1 REM ZX SPECTRUM Picture Convert
2 REM
3 REM By Michael Hadrup
4 REM
5 REM (C) 1986 MJH Software
6 REM
7 REM
10 RESTORE
20 X=&HF800
30 FORN=1000TO1160STEP10
40 C=0
50 FORM=0TO7
60 READA$
70 POKEX,VAL("&H"+A$)
80 C=C+PEEK(X)
90 X=X+1
100 NEXT
110 READA$
120 IFC<>VAL("&H"+A$)THENBEEP2:PRINT"Error in line ";N:STOP
130 NEXT
140 BEEP
150 BEEP
160 PRINT"No errors!!"
170 END
1000 DATA 1,4,6,D,2,7,A,E,39
1010 DATA 1,5,8,D,3,7,B,F,3F
1020 DATA F3,AF,D3,BF,3E,40,D3,BF,544
```

```
1030 DATA 21,0,C0,6,3,7E,24,D3,25F
1040 DATA BE,7C,E6,7,20,F7,7C,D6,490
1050 DATA 8,67,2C,20,F0,7C,C6,8,2F5
1060 DATA 67,10,EA,AF,D3,BF,3E,60,440
1070 DATA D3,BF,EB,1,0,3,1A,13,2AE
1080 DATA E6,BF,6F,CD,61,F8,7,7,448
1090 DATA 7,7,67,7D,F,F,F,CD,1EC
1100 DATA 61,F8,B4,2E,8,D3,BE,2D,401
1110 DATA 20,FB,B,78,B1,20,DF,FB,449
1120 DATA C9,E6,7,CB,7D,28,2,C6,3EE
1130 DATA 8,D9,6F,26,F8,7E,D9,C9,48E
1140 DATA FD,21,0,C0,11,0,1B,3E,248
1150 DATA FF,3F,CD,0,F9,D8,3E,41,45B
1160 DATA C3,A1,FB,0,0,0,0,0,25F
```

The machine code program

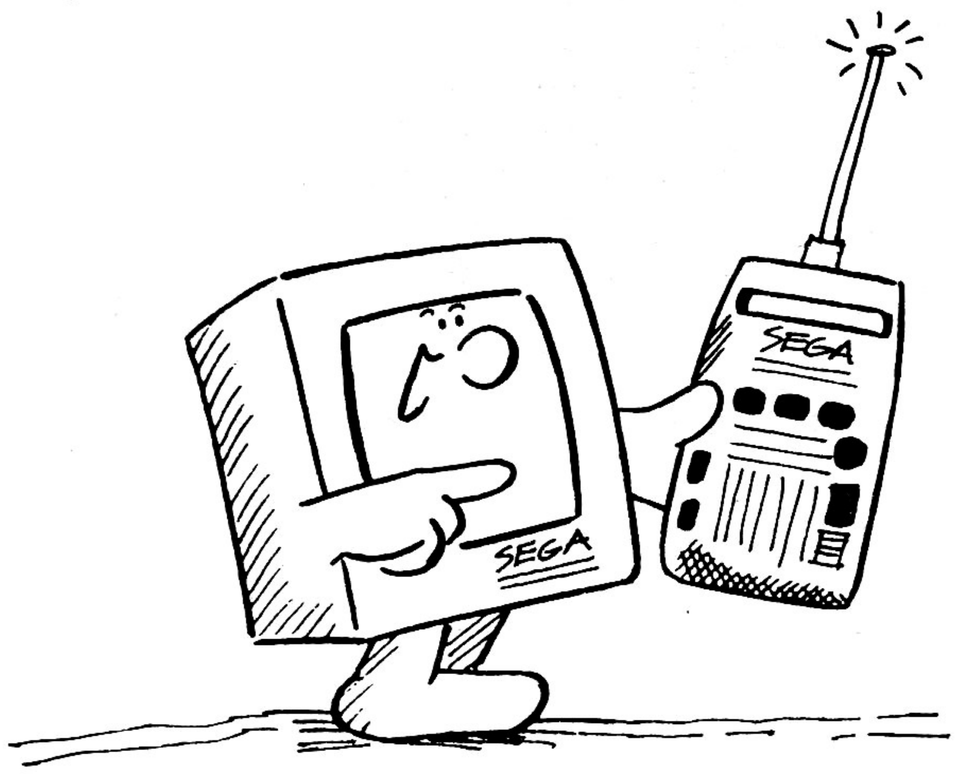
```
F810 F3          DI
F811 AF          XOR A
F812 D3BF        OUT (BF),A
F814 3E40        LD A,40
F816 D3BF        OUT (BF),A
F818 2100C0      LD HL,C000
F81B 0603        LD B,03
F81D 7E          LD A,(HL)
F81E 24          INC H
F81F D3BE        OUT (BE),A
F821 7C          LD A,H
F822 E607        AND 07
F824 20F7        JR NZ,F81D
F826 7C          LD A,H
F827 D608        SUB 08
F829 67          LD H,A
F82A 2C          INC L
F82B 20F0        JR NZ,F81D
F82D 7C          LD A,H
F82E C608        ADD A,08
F830 67          LD H,A
F831 10EA        DJNZ F81D
F833 AF          XOR A
F834 D3BF        OUT (BF),A
F836 3E60        LD A,60
F838 D3BF        OUT (BF),A
F83A EB          EX DE,HL
F83B 010003      LD BC,0300
F83E 1A          LD A,(DE)
F83F 13          INC DE
F840 E6BF        AND BF
F842 6F          LD L,A
F843 CD61F8      CALL F861
F846 07          RLCA
F847 07          RLCA
F848 07          RLCA
F849 07          RLCA
F84A 67          LD H,A
F84B 7D          LD A,L
F84C 0F          RRCA
F84D 0F          RRCA
F84E 0F          RRCA
F84F CD61F8      CALL F861
F852 B4          OR H
F853 2E08        LD L,08
F855 D3BE        OUT (BE),A
F857 2D          DEC L
F858 20FB        JR NZ,F855
F85A 0B          DEC BC
F85B 78          LD A,B
F85C B1          OR C
F85D 20DF        JR NZ,F83E
F85F FB          EI
F860 C9          RET
F861 E607        AND 07
F863 CB7D        BIT 7,L
F865 2802        JR Z,F869
F867 C608        ADD A,08
F869 D9          EXX
F86A 6F          LD L,A
F86B 26F8        LD H,F8
F86D 7E          LD A,(HL)
F86E D9          EXX
F86F C9          RET
F870 FD2100C0    LD IY,C000
F874 11001B      LD DE,1B00
F877 3EFF        LD A,FF
F879 3F          CCF
F87A CD00F9      CALL F900
F87D D8          RET C
F87E 3E41        LD A,41
F880 C3A1FB      JP FBA1
F883 00          NOP
F884 00          NOP
```

Fixing The Comset Command (Disk Basic)

By Michael Hadrup

This article should provide an answer for K.Nightingale's (of Whakatane) question about the comset command (Disk Basic). It has only four mistakes in this command! To fix the comset command run this program before using COMSET.

```
10000 READA$:A=VAL("&h"+A$):POKEA,PEEK(A)-13:GOTO10000
10010 DATA1F74,1F78,1F97,1F9A
```



Part 3
The SEGA Computer
THE SOUND GENERATOR

BY Brian Brown

The sound chip is a SN76489AN device. It requires 32 clock cycles for the transfer of data from the CPU to be latched internally. This involves the use of the Ready line being tied to the WAIT input of the Z80 CPU.

This means that when loading the sound generator chip with data the CPU is actually slowed down. The SG contains three programmable tone generators and a noise source, the output of each controlled by a programmable attenuator. The SG chip is port mapped at &H7F. The frequency and register is selected by a two-byte combination, while only one byte is necessary for attenuation control.

FREQUENCY SELECTION

To determine how to program the SG chip the following information is necessary;

Clock speed = 3.84Mhz
N = Clock speed / 32 * Required frequency)

where N is converted to a 10 digit binary number.

Thus, to generate a tone of 1000Hz;

N = 3840000 / 32 * 1000
= 120 (N is always rounded to an integer)

Now convert N to binary = 0001111000 (most significant bit first).

REGISTER SELECTION

To determine which sound register Table XX10 is used.

WRITING THE FREQUENCY AND REGISTER TO THE SGC

In the above example of a 1000Hz tone, N was derived into a 10 digit binary number of 0001111000. These 10 bits, along with the register code from Table XX10 are used to form the two bytes required to program the desired frequency and sound channel. Thus the format of the two bytes is;

Byte One: 1 + Register Code + last 4 bits of N
Byte Two: 00 + first 6 bits of N

Thus our example of a 1000Hz tone using register one;

Byte One = 10001000 (or &H88)
Byte Two + 00000111 (or &H07)

The tone is produced by outputting the two values to port &H7F, thus;

```
OUT &H7F,&H88 : OUT &H7F,&H07
```

will produce the desired result.

ATTENUATION CONTROL

Control of the programmable attenyators can be achieved by a single byte update. The format of this byte is as follows;

Single Byte = 1 + attenuation register + attenuation value

The attenuation register is three bits and is shown in Table XX11. The attenuation value is shown in Table XX12 and comprises four bits. Thus to attenuate tone register one to a value of 10db using BASIC.

Single Byte = 100010101 (&H95) so OUT &H7F,&H95

THE NOISE GENERATOR

Updating the noise register and attenuator requires a single byte transfer. This byte is 11100 + FB + SR.

FEEDBACK CONTROL (FB): If FB=1 then noise is "periodic" else if FB=1 then the noise is set to "white" noise.

SHIFT RATE (SR): Refer to Table XX13 for the values of the two SR bits.

ATTENUATION CONTROL OF NOISE REGISTER: this is the same as described earlier, only the register code is 111.

SAMPLE EXPLOSION: To generate an explosion, use "white noise" then slowly increase the attenuation from Obd to OFF. Thus the frequency control byte is;

11100 + 1 + 00 = 11100100 (or &HE4)

The attenuation bytes range from Odb to OFF thus the range is;

1111 + 0000 to 1111 = 1111 (or &HFO to &HFF)

thus the program in BASIC is;

```
10 OUT &H7F,&HE4 : FOR X = &HFO TO &HFF : OUT
     &H7F,X
20 FOR DE = 1 TO 20 : NEXT DE
30 NEXT X
```

CREATING MUSIC: Table XX14 is a BASIC program which allows the user to input a series of notes (up to 255) and then play them back. The program calculates the various bytes necessary to program the south generator chip.

TABLE XX10 REGISTER CODES

Register	*Binary Code*
Register 1	000
Register 2	010
Register 3	100
Register 4	110

TABLE XX11 ATTENUATOR CODES

Attenuator	*Binary Code*
Tone reg 1	001
Tone reg 2	011
Tone reg 3	101
Noise reg	111

TABLE XX12 ATTENUATION TABLE

Attenuation Value	*Binary Code*	*Attenuation Value*	*Binary Code*
0db	0000	2db	0001
4db	0010	6db	0011
8db	0100	10db	0101
12db	0110	14db	0111
16db	1000	18db	1001
20db	1010	22db	1011
24db	1100	26db	1101
28db	1110	OFF	1111

TABLE XX13 SHIFT RATE BITS

SR0	*SR1*	*Desired Frequency of Noise*
0	0	Clock/512
0	1	Clock/1024
1	0	Clock/2048
1	1	Frequency is that specified by Register 3

TABLE XX14 MUSIC PROGRAM & SOUND EDITOR

```
  5 PATTERNC#&HD0,"7884B4A4A4B48478"
 10 PN=&H7F:DIM X1(255),X2(255),X3)255),W(
    255),TZ(255)
 20 FLAG=0
 30 SCREEN1,1:CLS
 40 PRINT "Music Editor. By B.Brown. ";CHR$(&
    HD0);" 84"
 50 PRINT "-----------          ----------"
 60 PRINT:PRINT"Options"
 70 PRINT "1 — Play memory area"
 80 PRINT "2 — Create music"
 90 PRINT "÷ — Edit music array"
100 PRINT:PRINT "Select desired option:"
110 AA$=INKEY$:IF AA$=""THEN GOTO 110
120 IF AA$="1" THEN GOTO 820
130 IF AA$="2" THEN GOTO 900
140 IF AA$="3" THEN GOTO 1000
150 GOTO 110
200 REM INPUT ROUTINES
210 B1$="10000000"
220 PRINT"Freq (118 ~3500) of tone #";ZB;" ";:IN
    PUT FT:IF FT<118 OR FT>3500 THEN GOT
    O 220
230 BT=3840000/(32*FT)
240 DB=INT(BT+.5):GOSUB 430
640 SB$=SB$+MID$(SA$,LEN(SA$)+1—S,1)
650 NEXT S
660 RETURN
670 REM STRING TO DECIMAL
680 REM INPUT=GS$,OUTPUT=OB
690 OB=0
700 IFMID$(GS$,1,1)="1"THEN OB=OB+12B
710 IFMID$(GS1,2,1)="1"THEN BO=OB+64
720 IFMID$(GS$,3,1)="1"THEN OB=OB+32
730 IFMID$(GS1,4,1)="1"THEN OB=OB+16
740 IFMID$(GS$,5,1)="1"THEN OB=OB+8
750 IFMID$(GS$,6,1)="1"THEN OB=OB+4
760 IFMID$(GS$,7,1)="1"THEN OB=OB+2
770 IFMID$(GS$,8,1)="1"THEN OB=OB+1
780 RETURN
790 REM RESET SOUND CHANNELS
800 OUTPN,159:OUTPN,191:OUTPN,223
810 OUTPN,255:RETURN
820 REM PLAY MUSIC
830 CLS:PRINT"Playing music.":PRINT" ----------"
840 IF FLAG=0 THEN PRINT:PRINT"Music array i
    s empty.":GOSUB 1140:GOTO—30
850 FOR ZB=1 TO 255
860 N1=X1(ZB):N2=X2(ZB):N3=X3(ZB):ZC=
    W(ZB):IF N1=0 AND N2=0 and N3=0 THEN
    ZB=255:GOTO 880
870 GOSUB 390:SOUND 0
880 NEXT ZB
890 GOTO 30
900 REM Create music
910 CLS:PRINT "Create Music.":PRINT"----------
    -:PRINT:GOSUB 1140
920 INPUT"How many notes to play.";ZA
930 IF ZA>255 THEN GOTO 920
940 FOR ZB=1 TO ZA
950 GOSUB 700
960 X1(ZB)=N1:X2(ZB)=N2:X3(ZB)=N3;W(ZB)
    =ZC:TZ(ZB)=FT
970 NEXT:Z1(ZB)=0:X2(ZB)=0:X3(ZB)=0
980 GOSUM 1140:FLAG=1:GOTO 30
990 STOP
1000 REM Edit music
1010 CLS:PRINT "Edit Music.":PRINT"----------":P
     RINT:IF FLAG=0 THEN PRINT "Buffer is empt
     y.":GOSUB 1140:GOTO 30
1020 PRINT "Freq bytes can only be changed, not"
1030 PRINT "inserted. Use the ";CHR$(&H8E);" key
     to change a ":PRINT "tone, else ";CHR$(&H8F
     );" key to move to the next":PRINT "tone, and
     CR to abort."
1040 FOR ZB=1 TO 255
1050 PRINT "Tone ";ZB;" is ";TZ(ZB);"Hz"
1055 PRINT "Wait period is";W(ZB)
1060 TR$="":TR$INKEY$
1090 IF TR$chr$(30) THEN GOSUB 1150:GOTO 10
     50
1100 IF TR$=CHR$(29) THEN GOSUB 1140:NEX
     T
1110 IF TR$=CHR  (13) THEN 1130
1120 GOTO 1060
1130 GOSUB 1140:GOTO 30
1140 FOR DE=1 TO 200:NEXTDE:RETURN
1150 GOSUB 1140:GOSUB 200:X1(ZB)=N1:X2(Z
     B)=N2:X3(ZB)=N3:W(ZB)=ZC:TZ(ZB)=FT:
     RETURN
```

Part 4
The SEGA Computer
CASSETTE ROUTINES

BY Brian Brown

MAJOR ENTRY POINTS

The major entry points for the cassette are;

VERIFY	&H779F
LOAD	&H78D5
SAVE	7H7A40

PROGRAM FORMAT

The programs are saved in two stages. The first part is the Header section. This comprises sync bytes and the 16 character filename. The main program is saved next, this includes address's and the actual program, i.e., line numbers, etc.

VERIFY/LOAD

These routines are prefixed with a small routine which searches for the filename of the program. The filenames may be up to 16 characters long, and for the loading or verifying, the filename is stored at location &H82A3 onwards. The filename from header section of the tape is loaded, then compared with that stored in memory. If no filename was specified, the program jumps to the load main program section. If a filename was specified, and found to match with that read from tape, the program is loaded. If the program does not match, a jump is made to the skip portion of the program.

FILENAME STORAGE

Location &H82A2 is used as a Filename Found flag, and if zero then the next program found on the cassette is loaded, else a Filename Found flag, if zero then the program is loaded, otherwise skip is made. When saving a program, the filename is taken from the keyboard input buffer, corresponding to &H83A3 (up to 16 bytes). If the filename is less than 16 bytes, then the filename is padded with blanks.

ADDITIONAL INFO

Table XX15 lists the major entry points of all the cassette routines, and their function. Also listed are the subroutines which are called also. Table XX16 and XX17 are BASIC programs which load the Header and Program Bytes respectively to the video screen.

AUTOLOAD AND EXECUTE BASIC PROGRAMS

This may be achieved by poking a machine language routine into reserved memory. If the computer is then reset, the program will not be erased. The machine code routine calls the main entry point of LOAD, then calls (H6C37 which is the RUN entry point for BASIC programs. However, location &H82A2 which holds the Filename Found flag must be zeroed to indicate that the next program found must be loaded. Table XX18 illustrates how this may be achieved.

MERGING BASIC PROGRAMS

A program to merge two BASIC programs must use a machine code routine to save the RAM pointers in memory, call the Load routine in ROM, reset the pointers and call the load routine a second time. The program listed below is a combination of most of that which has already been covered. It must be noted however, that the second BASIC program's line numbers must be greater than the first or part of the program will be deleted.

```
10  SCREEN 1,1: CLS : PRINT "BASIC MERGER"
20  PRINT: PRINT "Load Mcode."
30  POKE &H8161,0 : FOR X=&HFFOO TO &HFF2
    F
35  REM Reserve memory space at top of memory
40  READ A: POKE X,A: NEXT
70  PRINT "Press PLAY to load first program."
80  POKE &H82A2,0: CALL &HFFOO
90  END
100 DATA &HCD,&HEF,&H78
110 DATA &H2A,&H62,&H81,&H2B,&H22,&H60,&
    H81
120 DATA &H3E,&H00,&H32,&HA2,&H82,&H21,&
    H1F
130 DATA &HFF,&HCD,&H6F,&H4A,&HCD,&HEF,
    &H78
140 DATA &H21,&HOO,&H98,&H22,&H60,&H81,&
    HC9
150 DATA 76,111,97,100,32,50,110,100,32,112
160 DATA 114,111,103,114,97,109,&HOD
```

In machine code the program is;

```
FFOO  CALL 78EF          (LOAD prog1)
      LD HL (8162)       (BASIC end pointer)
      DEC HL
      LD (8160),HL       (Store it into BASIC
                         start)
      LD A, 00
      LD (82A2),A        (Filename found flag)
      LD HL,FF1F
      CALL 4A6F          (Print text message)
      CALL 78EF          (Load prog2)
      LD HL,9800
      LD (8160),HL       (Set pointer to prog1)
      RET
FF1F  "load 2nd program" (Text message)
```

TABLE XX15 CASSETTE ROUTINES IN ROM

ROM Address (Hex)	*Function*
3A03	A Delay routine using the BC register
3A0F	Write sync bytes to tape
3A12	Write byte in A to tape
779F	Verifying Start
77F7	Skip
7822	Found
785D	Verifying End
788F	Verifying Error
78DF	Loading Start
78FD to 790E	Compare filenames
792B	Skip
7956	Found
7982	Load Program
799AA	Loading End
79E9	Tape Read Error
7A40	Saving Start
7A59 to 7A85	Save Filename
7A94	Save number of bytes
7AB9	Save Sync Bytes
7AD2	Save Program
7AED	Saving End
7B07	Write HL to tape
7B13	Pad Filename with Blanks

TABLE XX16 LOAD HEADER TO VIDEO SCREEN

```
10  SCREEN 1,1 : CLS : PRINT "Press Play to Load pr
    ogram." : B = 0
20  FOR X = &H78EF TO &H7923
30  POKE &HABEF+B,PEEK(X) : B = B + 1 : NEXT
    X
40  POKE &HA90B,&HD3
50  POKE &HA90C,&HBE
60  POKE &HA924,&HC9
70  CALL &HA8EF
80  GOTO 70
```

TABLE XX17 LOAD PROGRAM BYTES TO VIDEO

```
10  SCREEN 1,1 : CLS : PRINT "Press Play to Load P
    rogram."
20  FOR X = &HA000 TO &HA022
30  READ A : POKE X,A : NEXT
40  CALL &HA000
50  STOP
60  DATA &HF3,&HCD,&H00,&H3A,&HCD,&H06
70  DATA &H3A,&HFE,&H17,&H20,&HF5,&H2A
80  DATA &H60,&H81,&H06,&H00,&HCD,&H0A
90  DATA &H7A,&HD3,&HBE,&H3E,&H3F,&HC4
100 DATA &H48,&H24,&H23,&H1B,&H7A,&HB3
110 DATA &H30,&HF0,&HC3,&HA9,&H79
```

TABLE XX18 AUTO LOAD AND RUN BASIC PROGRAMS

```
10  SCREEN 1,1 : CLS : PRINT " Press Play to Load
    and Run Program."
20  DATA &HCD,&HD5,&H78,&HCD,&H37,&H6C
30  POKE &H8168,0
40  FOR X = &HF000 TO &HF005
50  READ A : POKE X,A : NEXT
60  POKE &H82A2,0
70  CALL &HF000
```

Part 5
The SEGA Computer
THE KEYBOARD AND JOYSTICKS

BY Brian Brown

The keyboard, joysticks, cassette and printer are all controlled by an interface chip (8255). This interface chip allows the connection of the devices to the CPU, and the transfer of information between them. The interface is programmed by the CPU, i.e., it is instructed on what to look for and what it must do. This process is normally transparent to the user, i.e., the user is unaware of the process's being executed.

THE KEYBOARD

The keyboard is arranged in a matrix layout of eight columns by eleven rows. Only one column may be activated at one time, and the columns are controlled by a decoder chip. The keyboard rows and connected to two different ports, only one can be read by the CPU at any time. An intersection (which occurs due to a keypress) between the column and the row of the matrix is detected by the CPU and is then interpreted to find out the actual key being pressed. Refer to Table XX20 for the key matrix layout.

THE 8255 PERIPHAL INTERFACE CHIP

This is a programmable chip, and allows the connection of the keyboard, cassette, printer and joysticks to the CPU. The PIA has three ports — A, B and C — and a control register. The information written to the control register will determine the status of each port (i.e., inputs or outputs). When the ports are used as outputs, the written data is latched or held internally until the next update. In the SEGA the following is to be noted;

Port A is input, mapped at &HDC, connected to X columns of key matrix.
Port B is input, mapped at &HDD, connected to X columns of key matrix.
Port C is output, mapped at &HDE, connected to Y columns of key matrix.
Control register is mapped at &HDF.

The data or words written to the control register to set up the specific ports as input or output are;

Bits	7	6	5	4	3	2	1	0
	1	0	0	1	0	0	1	0

Bit 4 = Controls A
Bit 3 = Controls C upper
Bit 1 = Controls B
Bit 0 = Controls C lower

thus the byte to initialise the PIA is &H92 or 146 decimal.

ADDRESSING THE KEY-MATRIX

The lower three bits (0,1,2) of Port C is used to address the Y columns of the key matrix. The output of Port C is applied to a 74LS145 BCD decoder, which provides a one out of eight output to activate only one Y column at a time. The status of the three lower Port C bits will determine which output of the decoding chip is activated. Table XX18 lists the combinations of these three bits and the resultant activated output of the decoder. Table XX20 lists the keyboard matrix.

SCANNING THE KEYBOARD USING MACHINE CODE

Table XX21 lists a BASIC program which pokes a machine code subroutine into memory. This routine initialises the PIA with &H92, then outputs a specified byte to Port C, thus selecting the desired Y column of the key matrix. This byte is specified in line &% of the program, and refer to Table XX19 for the value of the byte and it's appropriate column. It then loads the value of Ports A and B, storing them in &HA000 and &HA001 respectively, before returning to BASIC. By

checking the returned code from Port A or B, it is thus possible to search for a specified key press. Having assembled the routine into line 5 of the program, all data statements etc. can be delted from the final program. Table XX26 lists a program which scans the keyboard, and moves sprites etc., all using machine code.

MISCELLANEOUS CONNECTIONS

The remaining tables list the various connections of the SEGA and their appropriate function.

THE PRINTER PLOTTER

This relies on a single chip microprocessor, a 6805 up. Being a factory programmed device, it must be replaced in total, i.e., you haven't got access to the software controlling the 6805. Also note that the same mechanism is used by the ATARI, and COMMODORE printer plotters, and the spares are also the same, i.e., pens, etc. Some SHARP printers are also identical, so shop around for pens, paper, etc.

TABLE XX19 THE 74LS145 DECODER COMBINATIONS

PC2	*PC1*	*PC0*	*Y Column*	*Hex Byte*	*(outputted to &HDE)*
0	0	0	Y0	00	
0	0	1	Y1	01	
0	1	0	Y2	02	
0	1	1	Y3	03	
1	0	0	Y4	04	
1	0	1	Y5	05	
1	1	0	Y6	06	
1	1	1	Y7	07	

TABLE XX20 KEYBOARD MATRIX LAYOUT

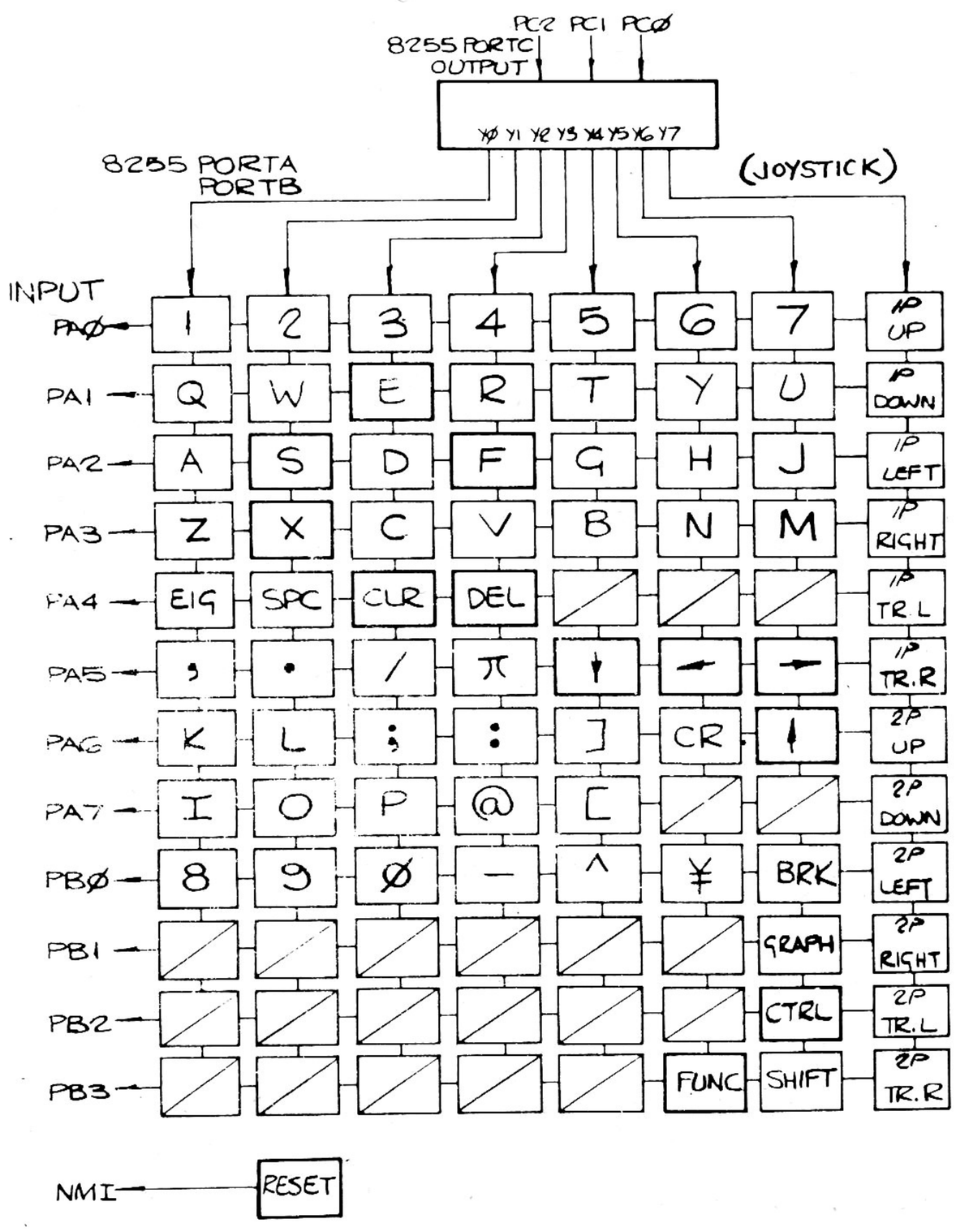

8255 PORTC OUTPUT
PC2 PC1 PCØ
YØ Y1 Y2 Y3 Y4 Y5 Y6 Y7
8255 PORTA
PORTB
(JOYSTICK)
INPUT
PAØ 1 2 3 4 5 6 7 1P UP
PA1 Q W E R T Y U 1P DOWN
PA2 A S D F G H J 1P LEFT
PA3 Z X C V B N M 1P RIGHT
PA4 EIG SPC CLR DEL 1P TR.L
PA5 ; • / π 1P TR.R
PA6 K L ; :] CR 2P UP
PA7 I O P @ [2P DOWN
PBØ 8 9 Ø − ^ ¥ BRK 2P LEFT
PB1 GRAPH 2P RIGHT
PB2 CTRL 2P TR.L
PB3 FUNC SHIFT 2P TR.R
NMI
RESET

TABLE XX21 BASIC KEY-SCAN PROGRAM

```
  5 REMAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        AAAAAAAAAAAAAAAAAAAAAAAAAAAA
  7 REM Line 5 has about 100 A's in it.
 10 SCREEN 1,1:CLS
 20 FOR X=&H9808 TO &H981F
 30 READ A:POKE X,A:NEXT X
 40 CALL &H9808
 50 PRINT"Port A =";PEEK(&HA001)" Port B =";P
    EEK(&HA001)
 60 GOTO 40
 65 DATA &HF3
 70 DATA &H3E,&H92,&HD3,&HDF
 75 DATA &H3E,&H00
 80 DATA &HD3,&HDE,&HDB,&HDC
 85 DATA &H32,&H00,&HA0,&HDB,&HDD
 90 DATA &H32,&H01,&HA0,&H3E,&H92
 95 DATA &HD3,&HDF,&HC9
100 REM Y0=00,Y1=01,Y2=02,Y3=03
110 REM Y4=04,Y5=05,Y6=06,Y7=07
120 REM Change the 2nd byte in line 75
130 REM to scan a different row.
```

TABLE XX22 JOYSTICK PIN CONNECTIONS

Pin Number	*Function*
1	Up
2	Down
3	Left
4	Right
5	No Connection
6	Left Fire
7	No Connection
8	Common
9	Right Fire

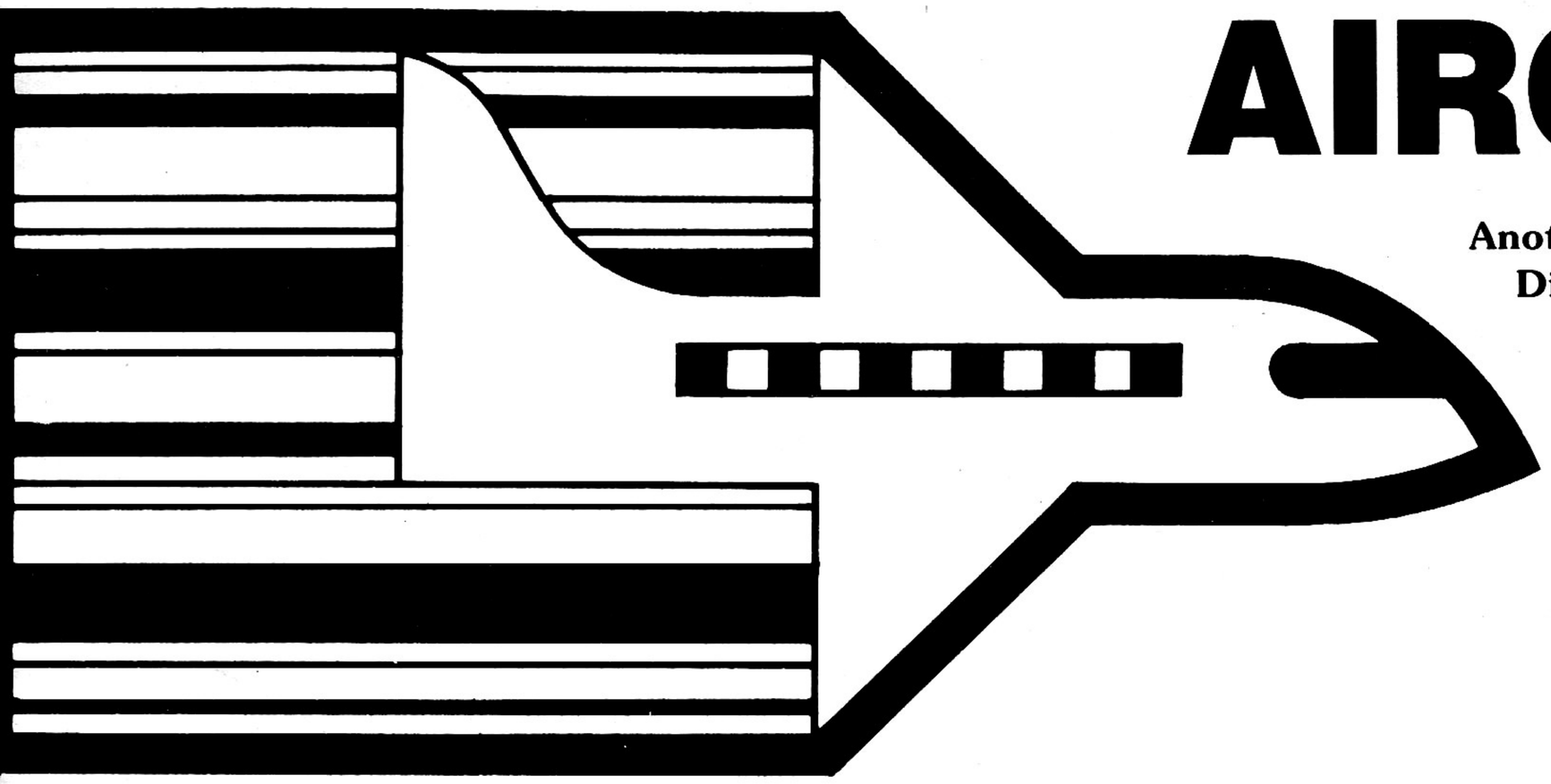
TABLE XX23 PRINTER PIN CONNECTIONS

Pin Number	*Function*
1	Fault
2	Busy
3	Data
4	Reset
5	Feed
6	Gnd
7	No Connection

TABLE XX24 PORT B & C CONNECTIONS

PB0	Key Matrix	PC0	Key Matrix
PB1	Key Matrix	PC1	Key Matrix
PB2	Key Matrix	PC2	Key Matrix
PB3	Key Matrix	PC3	Not Used
PB4	Not Used	PC4	Cassette Output
PB5	Fault (Printer)	PC5	Data (Printer)
PB6	Busy (Printer)	PC6	Reset (Printer)
PB7	Cassette Input	PC7	Feed (Printer)

cont on p55



AIRCRAFT

Another biggie from John Dowman. Difficult to beat the random firing of the enemy. Good graphics.

```
100 REM A I R C R A F T
110 REM By John Dowman 1986
120 REM ---------------------
130 REM
140 SCREEN 2,2:CLS:HS=0
150 COLOR ,1,(0,0)-(255,191),1
160 FOR Y=0 TO 13
170 LINE (0,Y*Y)-(255,Y*Y),4:NEXT Y
180 FOR X=1 TO 13
190 LINE (X*10,0)-(0,X*X),4:NEXT X
200 FOR X=14 TO 20
210 LINE (X*10,0)-(X*X-185,191),4:NEXT X
220 LINE (210,0)-(254,191),4
230 X=220
240 FOR Y=20 TO 18 STEP -1
250 LINE (X,0)-(255,Y*Y-300),4:X=X+10:NEXT Y
260 BLINE (47,50)-(191,60),,BF:COLOR15:CURSOR50,52:PRINTCHR$(16);"John Dowman Pr
esents..."
270 BLINE (100,150)-(210,163),,BF
280 X=0:Y=0:COLOR7:CURSOR102,155:PRINTCHR$(17);"AIRCRAFT";CHR$(16)
290 MAG 3
300 PATTERN S#0,"0000000000000001"
310 PATTERN S#1,"03070F0F0E000000"
320 PATTERN S#2,"0000001E2E6CF8F0"
330 PATTERN S#3,"E0C080000000000"
340 PATTERN S#4,"0000000000003F7F"
350 PATTERN S#5,"3E1C080000000000"
360 PATTERN S#6,"0000000818188000"
370 PATTERN S#7,"0000000000000000"
380 PATTERN S#8,"0000000000000000"
390 PATTERN S#9,"0F0E000303010000"
400 PATTERN S#10,"0000000000000020"
410 PATTERN S#11,"60E0E0E0E0E0E060"
420 PATTERN S#12,"0000000000007F7F"
430 PATTERN S#13,"3F1F0F0F07010000"
440 PATTERN S#14,"0000001E3E7CF8F0"
450 PATTERN S#15,"E0E0E0E0E0E0E060"
460 P=190:SOUND 4,2,15
470 FOR Q=0 TO 255 STEP 2
480 SPRITE 1,(Q,P),0,7
490 SPRITE 0,(Q,P),8,5
500 SPRITE 2,(Q,P),4,4
510 SPRITE 3,(Q,P+20),12,1:P=P-1.5
520 NEXT Q
530 FOR S=3500TO500 STEP -150:SOUND1,S,10:SOUND2,500+S,13:SOUND3,1000+S,15:NEXT
S:SOUND0:SOUND3,110,0
540 Q=0:P=0:S=0:BLINE(72,102)-(220,117),,BF
550 COLOR5:CURSOR75,106:PRINT"Press any key to begin.."
560 IF INKEY$="" THEN 560
570 BEEP
580 REM IF YOU HAVE BASIC Level III A THEN YOU CAN DELETE 4530-
```

```
590 REM AND PUT FOR LINE NUMBER 4530:-4350 RETURN
600 CLS:CURSOR70,100:PRINT "INSTRUCTIONS {Y/N}"
610 IF INKEY$="" THEN 610
620 IF INKEY$="Y" THEN GOSUB 4680:GOTO 650
630 IF INKEY$="N" THEN 650
640 GOTO 610
650 CLS
660 SCREEN 1,1:COLOR7,1:CLS
670 PATTERNC#48,"708898A8C88870"
680 Z$="INPUT YOUR INITIALS."
690 PRINT SPC(7);
700 FOR Z=1 TO LEN(Z$)
710 PRINT MID$(Z$,Z,1);
720 SOUND1,RND(1)*500+600,10:FORI=1 TO 5:NEXTI:SOUND0:NEXT Z
730 CURSOR0,5:INPUT N$
740 PRINT:PRINTSPC(3);"HELLO ";N$
750 FOR I=1 TO 500:NEXT I
760 N$=LEFT$(N$,3)
770 HS=0:SC=0:LI=20:MAG 1
780 PATTERNS#0,"01010163B6D5BEDF"
790 PATTERNS#1,"B7D6BDDEB7630100"
800 PATTERNS#2,"808080C66DAB7DFB"
810 PATTERNS#3,"ED6BBD7BEDC68000"
820 PATTERNS#4,"1E0B060203C2E27F"
830 PATTERNS#5,"775EEE4703020201"
840 PATTERNS#6,"78D06040C04347FE"
850 PATTERNS#7,"EE7A77E2C0404080"
860 PATTERNS#8,"071306010085771D"
870 PATTERNS#9,"7EE8CF2102000700"
880 PATTERNS#10,"940080600036CB68"
890 PATTERNS#11,"4EA1A6A960C04020"
900 PATTERNS#12,"0412050000816004"
910 PATTERNS#13,"4AC80B2600080701"
920 PATTERNS#14,"8400807000268B00"
930 PATTERNS#15,"88219401F0A04020"
940 PATTERNS#16,"0010020000814004"
950 PATTERNS#17,"480001A0008801A4"
960 PATTERNS#18,"8400000002208000"
970 PATTERNS#19,"0821000500800520"
980 PATTERNS#20,"80C0F8F0F8FF7FFE"
990 PATTERNS#21,"06FF7FF8F0F8C080"
1000 PATTERNS#22,"000000000000F05F
1010 PATTERNS#23,"5FF0"
1020 PATTERNS#24,"01061C23C080C220"
1030 PATTERNS#25,"190601"
1040 PATTERNS#26,"E01C03000000AA"
1050 PATTERNS#27,"5403FC"
1060 PATTERNS#28,"030303FF000000AA"
1070 PATTERNS#29,"0000FF"
1080 PATTERNS#30,"C03C03FF000100AA"
1090 PATTERNS#31,"0000FF"
1100 PATTERNS#32,"1F3164CE04E4A0E0"
1110 PATTERNS#33,"0000FF"
1120 PATTERNS#34,"0000808080808080"
1130 PATTERNS#35,"8080"
1140 PATTERNS#36,"0E070381C0E0FF87"
1150 PATTERNS#37,"87FFE0C08103070E"
1160 PATTERNS#38,"00F080C0E0F0FEF9"
1170 PATTERNS#39,"F9FEF0E0C080F0"
1180 PATTERNS#40,"000F0103070F7F9F"
1190 PATTERNS#41,"9F7F0F0703010F"
1200 PATTERNS#42,"70E0C0810307FFE1"
1210 PATTERNS#43,"E1FF070381C0E070"
1220 PATTERNS#44,"00000384FC57AAFC"
1230 PATTERNS#45,"0806"
1240 PATTERNS#46,"0000008080"
1250 PATTERNS#48,"FF01071F2152442A"
1260 PATTERNS#49,"110C07443F"
1270 PATTERNS#50,"FF80E0F827222127"
1280 PATTERNS#51,"24A8F012FC"
1290 SCREEN 2,2:COLOR,1,,5:CLS
```

```
1300 COLOR,8,(0,0)-(50,191)
1310 COLOR,11,(20,20)-(45,30)
1320 COLOR,11,(20,60)-(45,70)
1330 COLOR,11,(20,100)-(45,110)
1340 COLOR,11,(20,140)-(45,150)
1350 COLOR1:CURSOR24,22:PRINTCHR$(16);"MEN"
1360 CURSOR18,62:PRINT "HI~SC"
1370 CURSOR18,102:PRINT "SCORE"
1380 CURSOR20,142:PRINT "NAME"
1390 COLOR15:CURSOR20,80:PRINT HS
1400 CURSOR20,120:PRINT SC
1410 CURSOR18,40
1420 PRINTLI
1430 CURSOR25,160:PRINT N$
1440 X=60:Y=88:SOUND5,3,15
1450 FOR B=1 TO 10
1460 YY=INT(RND(1)*165+8):C=INT(RND(1)*13+2)
1470 FORXX=220TO60STEP-INT(RND(1)*10+5)
1480 SPRITE1,(XX,YY),48,C
1490 SPRITE2,(XX+16,YY),44,C
1500 IF STRIG(1)=1 THEN GOSUB 1660
1510 IF STICK(1)=1ANDY>15THENY=Y-8
1520 IF STICK(1)=5ANDY<160THENY=Y+8
1530 SPRITE0,(X,Y),36,4
1540 G=INT(RND(1)*25)+1
1550 ON G GOSUB 1720,1720,1730,1890,1840,1720,1720,1730,1890,1720,1720,1720,1720
,1890,1720,1840,1730,1720,1890,1730,1840,1720,1890,1780,1840
1560 NEXT XX
1570 NEXT B
1580 GOTO 2270
1590 LI=LI+1
1600 DATA 262,5,262,5,247,6,587,12,523,7,587,32
1610 RESTORE 1600:FOR Z=1 TO 6:READ P,DU:SOUND1,P,15:FOR I=1 TO DU:NEXT I:SOUNDO
:NEXT Z:SOUND5,3,15
1620 BLINE (10,35)-(50,50),,BF
1630 COLOR15:CURSOR18,38
1640 PRINT LI
1650 RETURN
1660 LINE (X+15,Y+8)-(240,Y+8),C
1670 SOUND1,1000,15
1680 BLINE (X+15,Y+8)-(240,Y+8)
1690 SOUNDO:SOUND5,3,15
1700 IF Y+8>YY AND Y+8<YY+14 THEN GOSUB 2130
1710 RETURN
1720 RETURN
1730 LINE (XX+8,YY+8)-(X+8,Y+30),C
1740 SOUND4,RND(1)*3,15
1750 BLINE (XX+8,YY+8)-(X+8,Y+30)
1760 SOUNDO:SOUND5,3,15
1770 RETURN
1780 LINE (XX+8,YY+8)-(X+8,Y+8),C
1790 SOUND4,RND(1)*3,15
1800 BLINE (XX+8,YY+8)-(X+8,Y+8)
1810 SOUNDO
1820 GOSUB 1980
1830 SOUND5,3,15:RETURN
1840 LINE (XX+8,YY+8)-(X+8,Y-5),C
1850 SOUND4,RND(1)*3,15
1860 BLINE (XX+8,YY+8)-(X+8,Y-5)
1870 SOUNDO
1880 SOUND5,3,15:RETURN
1890 Y1=INT(RND(1)*180)+3
1900 LINE (255,0)-(X+8,Y1),C
1910 LINE (255,191)-(X+8,Y1),C
1920 SOUND4,RND(1)*3,15
1930 BLINE (255,0)-(X+8,Y1)
1940 BLINE (255,191)-(X+8,Y1)
1950 SOUNDO
1960 IFY1>YANDY1<Y+16THENGOSUB1980
1970 SOUND5,3,15:RETURN
```

```
1980 SPRITE0,(X,Y),8,8
1990 FOR S=15 TO 11 STEP -3:SOUND4,2,S:FOR DY=1 TO 15:NEXT DY:NEXT S
2000 SPRITE0,(X,Y),12,8
2010 FOR S=10 TO 7 STEP -3:SOUND4,2,S:FOR DY=1 TO 15:NEXT DY:NEXT S
2020 SPRITE0,(X,Y),16,8
2030 FOR S=7 TO 0 STEP -3:SOUND4,2,S:FOR DY=1 TO 15:NEXT DY:NEXT S
2040 SPRITE0,(0,0),0,0
2050 SOUND0
2060 LI=LI-1:IF LI<=0 THEN 4300
2070 BLINE (15,35)-(50,50),,BF
2080 PRINT CHR$(16):COLOR15
2090 CURSOR18,40
2100 PRINT LI
2110 BEEP
2120 SOUND5,3,15:RETURN
2130 SPRITE1,(XX,YY),8,C
2140 FORS=15TO11STEP-3:SOUND4,2,S:FORDY=1TO15:NEXTDY:NEXTS
2150 SPRITE2,(0,0),0,0:SPRITE3,(0,0),0,0
2160 SPRITE1,(XX,YY),12,C
2170 FORS=10TO7STEP-3:SOUND4,2,S:FORDY=1TO15:NEXTDY:NEXTS
2180 SPRITE1,(XX,YY),16,9
2190 FORS=7TOOSTEP-3:SOUND4,2,S:FORDY=1TO15:NEXTDY:NEXTS
2200 SPRITE1,(0,0),0,0
2210 SOUND0
2220 XX=60:SC=SC+INT(RND(1)*25+75)
2230 IF SC>5000 AND SC<5100 THEN GOSUB 1590
2240 BLINE (15,115)-(50,130),,BF
2250 COLOR15:CURSOR18,118:PRINT SC
2260 SOUND5,3,15:RETURN
2270 SC=SC+INT(RND(1)*250+250)
2280 BLINE (15,115)-(50,130),,BF
2290 SOUND0:COLOR15:CURSOR18,118:PRINTSC
2300 CURSOR100,100:PRINT "NEXT ROUND"
2310 LINE (95,95)-(165,115),15,B
2320 FOR I=1 TO 500:NEXT I
2330 BLINE (90,90)-(170,120),,BF
2340 IFSC>5000ANDSC<5100 THENGOSUB 1590
2350 X=220:Y=88
2360 FOR B=1 TO 10
2370 T=INT(RND(1)*10+10):YY=INT(RND(1)*150+20):C=INT(RND(1)*13+2)
2380 FORXX=60 TO 230 STEP T
2390 SPRITE1,(XX,YY),20,C
2400 IF STRIG(1)=1 THEN GOSUB 2490
2410 IF STICK(1)=1ANDY>15 THEN Y=Y-8
2420 IF STICK(1)=5ANDY<160 THEN Y=Y+8
2430 SPRITE0,(X,Y),40,4
2440 G=INT(RND(1)*25)+1
2450 ON G GOSUB 1720,1720,1730,1890,1840,1720,1720,1730,1890,1720,1720,1720,1720
,1890,1720,1840,1730,1720,1890,1730,1840,1720,1890,1780,1840
2460 SOUND0:NEXT XX
2470 NEXT B
2480 GOTO 2590
2490 LINE (X+1,Y+8)-(60,Y+8),C
2500 SOUND1,1000,15
2510 BLINE (X+8,Y+8)-(60,Y+8)
2520 SOUND0
2530 IFY+8>YYANDY+8<YY+16THENGOSUB2560
2540 SOUND0
2550 RETURN
2560 GOSUB 2130
2570 XX=230
2580 RETURN
2590 SC=SC+INT(RND(1)*250+250)
2600 BLINE (15,115)-(50,130),,BF
2610 SOUND0:COLOR15:CURSOR18,118:PRINTSC
2620 CURSOR100,100:PRINT "NEXT ROUND"
2630 LINE (95,95)-(165,115),15,B
2640 FOR I=1 TO 500:NEXT I
2650 BLINE (90,90)-(170,120),,BF
2660 IFSC>5000ANDSC<5100THENGOSUB1590
```

```
2670 SCREEN 2,2:COLOR,1,,1:CLS:MAG1
2680 SPRITE0,(50,175),0,13
2690 SPRITE1,(50,5),20,12
2700 SPRITE2,(50,45),48,10:SPRITE3,(66,45),44,10
2710 SPRITE4,(34,85),24,15:SPRITE5,(50,84),28,15:SPRITE6,(66,84),32,15
2720 SPRITE7,(50,115),36,7
2730 SPRITE8,(50,150),4,8
2740 COLOR12:CURSOR100,10:PRINTCHR$(16);"= 200 pts":BEEP
2750 COLOR10:CURSOR100,50:PRINT "= 150 pts":BEEP
2760 COLOR15:CURSOR100,90:PRINT "= 300 pts":BEEP
2770 COLOR7:CURSOR100,120:PRINT "= 200 pts":BEEP
2780 COLOR8:CURSOR100,155:PRINT "= 400 pts":BEEP
2790 COLOR13:CURSOR100,180:PRINT "= Your Tank":BEEP
2800 RESTORE 2880
2810 FOR S=1 TO 15
2820 READ V
2830 SOUND1,V,15:FOR I=1 TO 50:NEXTI
2840 SOUND2,V,15:FOR I=1 TO 50:NEXTI
2850 SOUND3,V,15:FOR I=1 TO 50:NEXTI
2860 NEXT S
2870 SOUND0
2880 DATA 262,330,392,330,262,440,494,523,494,440,392,330,392,330,262
2890 SOUND3,110,0
2900 FOR I=1 TO 3
2910 T$=TIME$
2920 IF TIME$=T$ THEN 2920
2930 SOUND1,440,15
2940 FOR J=0 TO 10
2950 NEXT J
2960 SOUND0
2970 NEXT I
2980 T$=TIME$
2990 IF TIME$=T$ THEN 2990
3000 FOR I=15 TO 0 STEP -.2
3010 SOUND1,880,I
3020 NEXT I
3030 FORS=300TO3000STEP175:SOUND1,S,10:SOUND2,500+S,13:SOUND3,2000-(S/2),15:NEXT
S:FORI=1TO50:NEXTI:SOUND0
3040 FOR S=3500TO500 STEP-150:SOUND1,S,10:SOUND2,500+S,13:SOUND3,1000+S,15:NEXT
S:SOUND0
3050 FOR S=500 TO 3500 STEP150:SOUND1,ABS(S),10:SOUND2,500+S,13:SOUND3,1000+S,15
:NEXT S:SOUND0:X=118:Y=170
3060 SOUND3,110,0:FORI=1TO300:NEXTI
3070 SCREEN 2,2:COLOR,1,,1:CLS
3080 COLOR1,8,(0,0)-(255,10),1
3090 CURSOR20,2:PRINT "SCORE";SC
3100 CURSOR100,2:PRINT "HI~SC";HS
3110 CURSOR180,2:PRINT "MEN";LI
3120 CURSOR100,13:PRINT "NAME ";N$
3130 COLOR,15,(0,170)-(255,191)
3140 COLOR,8,(100,10)-(150,20)
3150 FOR B=1 TO 20
3160 G=INT(RND(1)*30)+1
3170 ON G GOSUB 3200,3310,3200,3420,3530,3640,3200,3310,3420,3530,3200,3640,3310
,3640,3530,3200,3310,3420,3530,3640,3200,3420,3310,3640,3530,3200,3200,3420,3310
,3640
3180 NEXT B
3190 SOUND0:GOTO 780
3200 P=200:C=12:YY=INT(RND(1)*50+35)
3210 FOR XX=10 TO 230 STEP 10
3220 SPRITE1,(XX,YY),20,C:SOUND0
3230 IF STRIG(1)=1 THEN GOSUB 3750
3240 IFSTICK(1)=3ANDX<230THENX=X+10
3250 IFSTICK(1)=7ANDX>10THENX=X-10
3260 SPRITE0,(X,Y),0,13:SOUND4,2,10
3270 H=INT(RND(1)*20)
3280 ONHGOSUB 3850,3850,3860,3850,3910,3850,3960,3850,3860,3910,3850,3850,3850,3
960,3960,3850,3860,3850,4050
3290 NEXT XX
3300 RETURN
```

```
3310 P=150:C=10:YY=INT(RND(1)*50+35)
3320 FOR XX=230 TO 10 STEP -5
3330 SPRITE1,(XX,YY),48,C:SPRITE2,(XX+16,YY),44,C
3340 IF STRIG(1)=1 THEN GOSUB 3750
3350 IFSTICK(1)=3ANDX<230THENX=X+10
3360 IFSTICK(1)=7ANDX>10THENX=X-10
3370 SPRITE0,(X,Y),0,13:SOUND4,2,10
3380 H=INT(RND(1)*20)
3390 ONHGOSUB 3850,3850,3860,3850,3910,3850,3960,3850,3860,3910,3850,3850,3850,3
960,3960,3850,3860,3850,4050
3400 NEXT XX
3410 RETURN
3420 P=300:C=15:YY=INT(RND(1)*50+35)
3430 FOR XX=210 TO 10 STEP -15
3440 SPRITE1,(XX,YY),24,C:SPRITE2,(XX+16,YY-1),28,C:SPRITE3,(XX+32,YY-1),32,C
3450 IF STRIG(1)=1 THEN GOSUB 3750
3460 IFSTICK(1)=3ANDX<230THENX=X+10
3470 IFSTICK(1)=7ANDX>10THENX=X-10
3480 SPRITE0,(X,Y),0,13:SOUND4,2,10
3490 H=INT(RND(1)*20)
3500 ONHGOSUB 3850,3850,3860,3850,3910,3850,3960,3850,3860,3910,3850,3850,3850,3
960,3960,3850,3860,3850,4050
3510 NEXT XX
3520 RETURN
3530 P=200:C=7:YY=INT(RND(1)*50+35)
3540 FOR XX=10 TO 230 STEP 10
3550 SPRITE1,(XX,YY),36,C
3560 IF STRIG(1)=1 THEN GOSUB 3750
3570 IFSTICK(1)=3ANDX<230THENX=X+10
3580 IFSTICK(1)=7ANDX>10THENX=X-10
3590 SPRITE0,(X,Y),0,13:SOUND4,2,10
3600 H=INT(RND(1)*20)
3610 ONHGOSUB 3850,3850,3860,3850,3910,3850,3960,3850,3860,3910,3850,3850,3850,3
960,3960,3850,3860,3850,4050
3620 NEXT XX
3630 RETURN
3640 P=400:C=8:XX=INT(RND(1)*200+20)
3650 FOR YY=35 TO 180 STEP 20
3660 SPRITE1,(XX,YY),4,C
3670 IF STRIG(1)=1 THEN GOSUB 3750
3680 IFSTICK(1)=3ANDX<230THENX=X+10
3690 IFSTICK(1)=7ANDX>10THENX=X-10
3700 SPRITE0,(X,Y),0,13:SOUND4,2,10
3710 H=INT(RND(1)*20)
3720 ONHGOSUB 3850,3850,3860,3850,3910,3850,3960,3850,3860,3910,3850,3850,3850,3
960,3960,3850,3860,3850,4050
3730 NEXT YY
3740 RETURN
3750 LINE (X+8,Y+1)-(X+8,30),C
3760 SOUND1,1000,15
3770 BLINE (X+8,Y+1)-(X+8,30)
3780 SOUND0
3790 IF C=12ANDX+8>XXANDX+8<XX+16 THENGOSUB 4110
3800 IF C=10ANDX+8>XXANDX+8<XX+31 THEN GOSUB 4110
3810 IF C=15ANDX+8>XXANDX+8<XX+45 THEN GOSUB 4110
3820 IFC=7ANDX+8>XXANDX+8<XX+16THENGOSUB4110
3830 IFC=8ANDX+8>XXANDX+8<XX+16THENGOSUB4110
3840 RETURN
3850 RETURN
3860 LINE (XX+8,YY+8)-(X-8,Y+8),C
3870 SOUND4,RND(1)*3,15
3880 BLINE (XX+8,YY+8)-(X-8,Y+8)
3890 SOUND0
3900 RETURN
3910 LINE (XX+8,YY+8)-(X+17,Y+8),C
3920 SOUND4,RND(1)*3,15
3930 BLINE (XX+8,YY+8)-(X+17,Y+8)
3940 SOUND0
3950 RETURN
3960 R=INT(RND(1)*255)
```

```
3970 LINE (0,20)-(R,Y+8),C
3980 LINE (255,20)-(R,Y+8),C
3990 SOUND4,RND(1)*3,15
4000 BLINE (0,20)-(R,Y+8)
4010 BLINE (255,20)-(R,Y+8)
4020 SOUND0
4030 IF R>XANDR<X+16 THEN GOSUB4560
4040 RETURN
4050 LINE (XX+8,YY+8)-(X+8,Y+8),C
4060 SOUND4,RND(1)*3,15
4070 BLINE (XX+8,YY+8)-(X+8,Y+8)
4080 SOUND0
4090 GOSUB 4560
4100 RETURN
4110 SPRITE1,(XX,YY),8,C
4120 FOR S=15 TO 11 STEP -3:SOUND4,1,S:FOR DY=1 TO 15:NEXT DY:NEXT S
4130 SPRITE2,(0,0),0,0:SPRITE3,(0,0),0,0:SPRITE1,(XX,YY),12,C
4140 FOR S=10 TO 7 STEP -3:SOUND4,1,S:FOR DY=1 TO 15:NEXT DY:NEXT S
4150 SPRITE1,(XX,YY),16,C
4160 FOR S=7 TO 0 STEP -3:SOUND4,1,S:FOR DY=1 TO 15:NEXT DY:NEXT S
4170 SPRITE1,(0,0),0,0
4180 SOUND0
4190 IF C=12 THEN XX=230
4200 IF C=10 THEN XX=10
4210 IF C=15 THEN XX=10
4220 IF C=7 THEN XX=230
4230 IF C=8 THEN YY=180
4240 SC=SC+P
4250 BLINE (53,0)-(93,10),,BF
4260 COLOR1:CURSOR55,2:PRINT SC
4270 IFSC>5000ANDSC<5500THENGOSUB1590:BLINE (20,30)-(40,50),,BF
4280 BLINE (200,0)-(220,10),,BF:COLOR1:CURSOR203,2:PRINT LI
4290 RETURN
4300 SCREEN 2,2:COLOR,1,,1:CLS
4310 X$="GAME OVER!":I=40
4320 SOUND0
4330 COLOR9,1
4340 FOR O=1 TO LEN(X$)
4350 Y$=MID$(X$,O,1)
4360 LINE (125,96)-(I+6,132),9
4370 BLINE (125,96)-(I+6,132),9
4380 I=I+14
4390 FOR Z=15 TO 1 STEP -2
4400 SOUND1,320,Z:SOUND0:SOUND1,330,Z
4410 NEXT Z
4420 CURSORI,133:PRINT CHR$(17);Y$
4430 LINE (I-8,146)-(I+15,146),9,BF
4440 SOUND0
4450 NEXT O
4460 FOR AD=15 TO 1 STEP -1
4470 A=INT(RND(1)*200)
4480 SOUND1,310,AD:SOUND1,360,AD:SOUND1,410,AD
4490 NEXT AD
4500 SOUND0:FOR I=1 TO 500:NEXT I
4510 IF SC>HS THEN HS=SC
4520 CURSOR75,50:PRINTCHR$(16);"PLAY AGAIN? [Y/N]"
4530 IF INKEY$="" THEN 4530
4540 IF INKEY$="Y" THEN SC=0:LI=20:SCREEN 1,1:CLS:INPUT "INITIALS:-";N$:N$=LEFT$
(N$,3):SCREEN 2,2:GOTO 780
4550 END
4560 SPRITE0,(X,Y),8,9
4570 FOR S=15 TO 11 STEP -3:SOUND4,1,S:FOR DY=1 TO 15:NEXT DY:NEXT S
4580 SPRITE0,(X,Y),12,9
4590 FOR S=10 TO 7 STEP -3:SOUND4,1,S:FOR DY=1 TO 15:NEXT DY:NEXT S
4600 SPRITE0,(X,Y),16,9
4610 FOR S=7 TO 0 STEP -3:SOUND4,1,S:FOR DY=1 TO 15:NEXT DY:NEXT S
4620 SPRITE0,(0,0),0,0
4630 SOUND0
4640 LI=LI-1:IF LI<=0 THEN 4300
4650 BLINE (200,0)-(220,10),,BF
```

cont on p55

AMAZE

By R. D. & A. A. Morgan

As this program is written purely in BASIC it tends to be a little slow in building up the screens, however it is a good exercise in strategy.

```
100 REM A M A Z E
110 REM TAPE VERSION
120 REM By R D & A A Morgan
130 REM ----------------------
140 REM
150 GOSUB 1440
160 MAG 0:SCREEN 2,2:CLS
170 COLOR 4,1,(0,0)-(255,191),0
180 DIM A$(38,22):T=0
190 PATTERN S#0,"0055497F223E1400"
200 PATTERN S#15,"081C2A7F63351C08"
210 Q=0:RO=1:EN$="N":O=0:OQ=0:OF=0
220 REM ----------------------------
230 CURSOR 15,182:COLOR 12,1:PRINT "A M A Z E";
240 IF G=1 THEN PRINT "                COMPUTER OPPONENT":GOTO 260
250 PRINT "                  PLAYER OPPONENT"
260 COLOR 4,1:CURSOR 10,2:PRINT "HI-SCORE=";:COLOR 12,1:PRINT H;" ";NM$:CURSOR 1
65,2:COLOR 4,1:PRINT "LEVEL=";:COLOR 12,1:PRINT RO:COLOR4,1:CURSOR10,12:PRINT "T
ODAYS-HI=";:COLOR12,1:PRINT T
270 GOSUB 520:CURSOR 165,12:COLOR4,1:PRINT "FOOD=":CURSOR85,22:PRINT "SCORE="
280 OQ=Q:OF=F
290 CURSOR130,22:COLOR 12,1:BLINE(125,22)-(195,30),,BF:PRINT Q:CURSOR 201,12:BLI
NE(201,12)-(250,20),,BF:PRINT F
300 IF STICK(1) <> 0 THEN GOSUB 750
310 IF F<1 THEN GOTO 370
320 IF G=1 THEN GOSUB 820
330 IF G=2 THEN Z$=INKEY$:IF Z$<>"" THEN GOSUB 820
340 IF EX=MX AND EY=MY THEN GOTO 390
350 IF OQ<>Q OR OF<>F THEN GOTO 280
360 GOTO 300
370 RO=RO+1:IF Q>T THEN T=Q
380 CLS:GOTO 230
390 GOSUB 630:GOSUB 670:SOUND 5,1,13:FOR A=1 TO 400:NEXT A:SOUND 0:CLS
400 IF Q>T THEN T=Q:IF T>H THEN H=T:BEEP 2
410 CURSOR  70,85:COLOR 15,1:PRINT "SCORE=";Q;" HI-SCORE=";H
420 CURSOR 110,95:COLOR 10,1:PRINT "GAME OVER"
430 CURSOR 70,105:PRINT "LEFT BUTTON - PLAY AGAIN"
440 CURSOR 82,115:PRINT "RIGHT BUTTON - FINISH"
450 MAG 3
460 GOSUB 1330:MX=5:MY=11:EX=34:EY=11:GOSUB 630:GOSUB 670
470 IF STRIG(1)=1 THEN SOUND 0:MAG 0:CLS:GOTO 210
480 GOSUB 1380:MX=34:MY=11:EX=5:EY=11:GOSUB 630:GOSUB 670
490 IF STRIG(1)=2 THEN SOUND 0:END
500 SOUND 5,3,10:GOTO 460
510 REM SET UP AND DRAW MAZE
520 F=0:FOR AX = 3 TO 36 STEP G :FOR AY = 4 TO 20
530 X=INT(AX*6.5):Y=INT(AY*8.3)
540 R=(RND(1)-RO/50)
550 A$(AX,AY)=" "
560 IF R>.73 THEN A$(AX,AY)="W":CURSORX,Y:COLOR4,1:PRINT "È"
570 IF R<.06 THEN A$(AX,AY)="F":F=F+1:CURSORX,Y:COLOR12,1:PRINT "Á"
580 NEXT AY:NEXT AX:SOUND 0:BEEP 2
590 MY=12:MX=2:GOSUB 630:EX=37:EY=12:GOSUB 670
600 RETURN
610 REM ----------------------------
620 REM DRAW GOODIE
630 X=INT(MX*6.5-1.3)
640 Y=INT(MY*8.3-1.5)
650 SPRITE 1,(X,Y),15,15:RETURN
660 REM DRAW ENEMY
```

```
670 X=INT(EX*6.5-1.3)
680 Y=INT(EY*8.3-1.5)
690 SPRITE 0,(X,Y),0,9:RETURN
700 REM DRAW OUT FOOD
710 X=INT(SX*6.5)
720 Y=INT(SY*8.3)
730 CURSORX,Y:COLOR1,1:PRINT "Ä":RETURN
740 REM ----------------------------
750 SX=MX:SY=MY:SS=STICK(1)
760 GOSUB 1190
770 IF A$(SX,SY)="F" THEN F=F-1:A$(SX,SY)=" ":GOSUB 710:Q=Q+10*RO:SOUND 3,415,15
:SOUND 3,659,15:SOUND 0
780 IF A$(SX,SY)="W" THEN BEEP:Q=Q-5:GOTO 800
790 MX=SX:MY=SY:GOSUB 630
800 RETURN
810 REM ----------------------------
820 SX=EX:SY=EY:SS=0
830 IF G=1 THEN 900
840 IF Z$=CHR$(13) THEN GOSUB 1680:RETURN
850 IF Z$=CHR$(29) THEN SS=7
860 IF Z$=CHR$(30) THEN SS=1
870 IF Z$=CHR$(31) THEN SS=5
880 IF Z$=CHR$(28) THEN SS=3
890 GOTO 980
900 IF MX<EX AND MY<EY THEN SS=8:GOTO 980
910 IF MX=EX AND MY<EY THEN SS=1:GOTO 980
920 IF MX>EX AND MY<EY THEN SS=2:GOTO 980
930 IF MX<EX AND MY=EY THEN SS=7:GOTO 980
940 IF MX>EX AND MY=EY THEN SS=3:GOTO 980
950 IF MX<EX AND MY>EY THEN SS=6:GOTO 980
960 IF MX=EX AND MY>EY THEN SS=5:GOTO 980
970 IF MX>EX AND MY>EY THEN SS=4
980 GOSUB 1190
990 IF G=1 AND A$(SX,SY)="W" THEN GOTO 1030
1000 IF G=2 AND A$(SX,SY)="W" THEN RETURN
1010 EX=SX:EY=SY:GOSUB 670:SOUND 5,3,13
1020 RETURN
1030 SX=EX:SY=EY
1040 IF MY<EY THEN SS=1:GOTO 1070
1050 IF MY>EY THEN SS=5:GOTO 1070
1060 GOTO 1100
1070 GOSUB 1190
1080 IF A$(SX,SY)="W" THEN GOTO 1100
1090 GOTO 1010
1100 SX=EX:SY=EY
1110 IF MX<EX THEN SS=7:GOTO 1140
1120 IF MX>EX THEN SS=3:GOTO 1140
1130 SOUND 0:GOTO 1020
1140 GOSUB 1190
1150 IF A$(SX,SY)="W" THEN SOUND 0:GOTO 1020
1160 GOTO 1010
1170 REM ----------------------------
1180 REM COMMON CHECK ROUTINE
1190 ON SS GOSUB 1250,1260,1270,1280,1290,1300,1310,1320
1200 IF SX > 37 THEN SX=37
1210 IF SX < 2 THEN SX=2
1220 IF SY < 2 THEN SY=2
1230 IF SY > 22 THEN SY=22
1240 RETURN
1250 SY=SY-1:RETURN
1260 SY=SY-1:SX=SX+1:RETURN
1270 SX=SX+1:RETURN
1280 SX=SX+1:SY=SY+1:RETURN
1290 SY=SY+1:RETURN
1300 SY=SY+1:SX=SX-1:RETURN
1310 SX=SX-1:RETURN
1320 SX=SX-1:SY=SY-1:RETURN
1330 SOUND 3,131,12:SOUND 3,139,12:SOUND 3,147,12:SOUND 3,156,12:SOUND 3,165,12:
SOUND 3,175,12:SOUND 3,185,12:SOUND 3,196,12:SOUND 3,208,12:SOUND 3,220,12
1340 SOUND 3,233,12:SOUND 3,247,12
1350 RETURN
```

```
1360 IF MX<EX THEN SS=7:GOTO 1070
1370 IF MX>EX THEN SS=3:GOTO 1070
1380 SOUND 3,233,11:SOUND 3,220,11
1390 SOUND 3,208,11:SOUND 3,196,11
1400 SOUND 3,185,11:SOUND 3,175,11
1410 SOUND 3,165,11:SOUND 3,156,11
1420 SOUND 3,147,11:SOUND 3,139,11
1430 SOUND 3,131,11:RETURN
1440 CLS:COLOR 15,1
1450 PRINT "            ** A M A Z E **             "
1460 PRINT "-A HIDE AND SEEK TREASURE HUNT GAME -"
1470 PRINT "       DESIGNED BY R.D.MORGAN           "
1480 PRINT :PRINT :PRINT
1490 PRINT "     A RANDOM PATTERN OF BLOCKS BEING IMPENETRABLE WALLS AND COLUMNS
 IS GEN-ERATED WITH A SCATTERING OF TREASURE  OR FOOD WHICH MUST BE COLLECTED TO
    GAIN POINTS."
1500 PRINT :PRINT "    WHEN THE SCREEN IS FILLED A SPRITE-YOU- APPEARS ON THE LE
FT HAND EDGE.  THIS IS CONTROLLED BY THE JOYSTICK."
1510 PRINT "    ON THE RIGHT HAND EDGE ANOTHER    SPRITE APPEARS.  THIS IS YOUR
ENEMY   WHICH WILL ALWAYS MOVE DIRECTLY TOWARDYOU IN ORDER TO KILL YOU !"
1520 PRINT :PRINT :PRINT  "PRESS ANY KEY TO CONTINUE."
1530 A$=INKEY$: IF A$="" THEN 1530
1540 CLS
1550 PRINT "    YOU MUST CONTROL YOUR ENEMY'S     DIRECTION OF MOVEMENT BY YOUR
OWN     MOVEMENTS IN SUCH A WAY THAT HE IS    TRAPPED BEHIND A WALL AND YOU CAN
GO  AND COLLECT FOOD IN SAFETY."
1560 PRINT "    ONE FORGETFUL MOMENT HOWEVER AND  HE WILL ESCAPE AND BE AFTER YO
U AGAIN!"
1570 PRINT :PRINT "    WHEN YOU GET THROUGH THE ROUND    ANOTHER SCREEN WILL BE
GENERATED WITH FEWER WALLS BUT MORE FOOD - AND HIGHERPOINTS."
1580 G=2
1590 PRINT:PRINT:PRINT :PRINT "     PLAY AGAINST COMPUTER? Y/N ";
1600 A$=INKEY$:IF A$="" THEN 1600
1610 PRINT A$:IF A$="Y" THEN G=1
1620 FOR MN=1 TO 100:NEXT MN
1630 PRINT:PRINT :PRINT "     DO YOU WISH TO PLAY?  Y/N ";
1640 A$=INKEY$:IF A$="Y" THEN 1670
1650 IF A$="N" THEN CLS:END
1660 GOTO 1640
1670 PRINT "YES":RETURN
1680 IF MX=EX OR MY=EY THEN GOTO 1700
1690 BEEP:BEEP:BEEP:RETURN
1700 IF MX=EX AND MY=EY THEN RETURN
1710 SG=SGN(MY-EY):IF MX=EX THEN FOR EY=EY TO (MY-3*SG) STEP SG:GOSUB 670:NEXT E
Y:RETURN
1720 SG=SGN(MX-EX):FOR EX=EX TO (MX-3*SG) STEP SG:GOSUB 67C:NEXT EX
1730 RETURN
```


NEW SOFTWARE

::: CASSETTE :::

MOONBASE ALPHA

From Mike Boyd of
"Sir Roderick's Quest" fame

Available 1st March

$30.25

DISK GAMES

NOW AVAILABLE!!!

Goldminer
Cyco
Sprite Editor } **$43.95**

Froggy
Cubit } **$43.95**

Or all five on one disk for $60.00

HACKERS CORNER

THE HIDDEN Z80 OPCODES SCANDLE

By MICHAEL HOWARD

Last time I gave you the complete set of Z80 instructions, the reason for this was simple, not many people had an easy-to-use table of the instructions. Most people who are into the Z80 have a copy of "Programming the Z80" by Rodney Zaks, and as we all know, the Z80 table in that is diabolical!

This month, as promised, I will show you the HIDDEN OPCODES! These 98 "hidden" opcodes, have been known for quite a few years now, so I take absolutely no credit for revealing them. Some games writers use them usually for convenience, or in protection systems to confuse the unwary hacker!

So what are these extra commands, and where have they come from? The majority of them are concerned with the IX and IY index registers, but there is one shift instruction that appears.

If you take a look at a table of the Rotate and Shift instructions, all but one of them has a left and right version:

RLC	RRC
RL	RR
SLA	SRA
RLD	RRD
	SRL

As you can see, the final one, SRL (Shift Right Logical) has no left hand counterpart — which should be SLL (Shift Left Logical), and there is a gap in the logical sequence of hex codes where these instructions should be from &CB &30 to &CB &37.

Experimenting with these opcodes shows that the SLL instructions are in fact recognised, and a logical shift to the left does take place. But it does not follow the normal rules of shifting! When a SRL instruction takes place, all the bits jump one bit to the Right: bit 0 goes into the Carry flag; and a 0 is placed in bit 7.

SLL would be expected to do the same, and it does, to a certain extent! It shifts the register to the left by one bit; bit 7 goes into the Carry flag, and a 1, NOT A 0, goes into bit 0. All the flags are set/reset exactly as SRL. So as you see, the only real difference is that a 1 is placed in the byte instead of an 0. Why not use the instruction? The opcodes are at the end of the article.

But let's get back to the majority of these hidden codes, which relate to the IX and IY index registers. If you study the opcode for any index register, you'll see that it is in fact identical to using the HL register pair, the only difference is that you add &DD (for IX? or &FD (for IY) at the start of the instruction, and then include an offset byte.

E.g.	ADD A(HL)	86	ADD A,(IX+2)	DD 86 02
	DEC (HL)	35	DEC (IX+5)	DD 35 02
	LD (nnnn),HL	22 nn nn	LD (nnnn),IY	FD 22 nn nn

However there are some gaps in the sequence of codes for the index register instructions. Take, as an example, LD A,H (hex code &7C). There is no published IX or IY equivalent, whose code would be DD 7C or FD 7C. If you experiment with these codes, there is in fact an effect! If you try FD 7C, the top 8 bits of the IY register are leaded into the Accumulator (A-register). If you try DEC L with a DD at the start, the lower 8 bits of the IX register are DECremented!

All these new instructions relate to the individual halves of the index registers (don't forget, they are 16 bit registers, not 8 bit!), and as their opcodes relate directly to the H and L registers, they are sometimes called the XH and XL for the High and Low halves of the IX register, and YH and YL for the High and Low halves of the IY register.

So, suddenly you have four extra 8 bit registers to play with! IXH, IXL, IYH and IYL. Great!

Although these new instructions are not documented by Zilog in their Z80 documentation, I have yet to come across a Z80 that does not accept them, so have a bit of fun with them, and tell me if you find any interesting ideas through using them.

The Hidden Opcodes

ADC A,XH	DD 8C	LD XH,A	DD 67
ADC A,XL	DD 8D	LD XH,B	DD 60
ADD A,XH	DD 84	LD XH,C	DD 61
ADD A,XL	DD 85	LD XH,D	DD 62
AND XH	DD A4	LD XH,E	DD 63
AND XL	DD A5	LD XL,A	DD 6F
CP XH	DD BC	LD XL,B	DD 68
CP XL	DD BD	LD XL,C	DD 69
DEC XH	DD 25	LD XL,D	DD 6A
DEC XL	DD 2D	LD XL,E	DD 6B
INC XH	DD 24	LD XL,XH	DD 6C
INC XL	DD 2C	LD XL,nn	DD 2E nn
LD A,XH	DD 7C	LD XH,XL	DD 65
LD A,XL	DD 7D	LD XH,nn	DD 26 nn
LD B,XH	DD 44	OR XH	DD B4
LD B,XL	DD 45	OR XL	DD B5
LD C,XH	DD 4C	SBC A,XH	DD 9C
LD C,XL	DD 4D	SBC A,XL	DD 9D

LD D,XH	DD 54	SUB XH	DD 94
LD D,XL	DD 55	SUB XL	DD 95
LD E,XH	DD 5C	XOR XH	DD AC
LD E,XL	DD 5D	XOR XL	DD AD

For the IY commands, replace the DD with FD

SLL (HL)	CB 36	SLL C	CB 31
SLL (IX + n)	DD CB n 36	SLL D	CB 32
SLL (IY + n)	FD CB n 36	SLL E	CB 33
SLL A	CB 37	SLL H	CB 34
SLL B	CB 30	SLL L	CB 35

```
10 REM MR MOO
20 REM A graphics demo
30 REM Brett Skellon
40 REM
50 SCREEN 2,2:COLOR15,15,,1:CLS
60 CIRCLE(100,70),17,6,.9,0,1
70 CIRCLE(93,67),3,6,1,0,1,BF
80 CIRCLE(107,67),3,6,1,0,1,BF
90 CIRCLE(100,70),10,6,.9,.1,.4
100 CIRCLE(100,50),20,6,1,.4,.12
110 CIRCLE(92,50),7,6,1.1,.32,.12
120 CIRCLE(108,50),7,6,1.1,.32,.22
130 CIRCLE(92,50),3,1,1,0,1,BF
140 CIRCLE(108,50),3,1,1,0,1,BF
150 CIRCLE(75,55),15,6,1,.15,.84
160 CIRCLE(125,55),15,6,1,.67,.35
170 CIRCLE(75,55),6,6,1,.15,.84
180 CIRCLE(125,55),6,6,1,.67,.37
190 CIRCLE(80,29),10,11,1,.85,.1
200 CIRCLE(80,29),15,11,.7,.83,.07
210 CIRCLE(120,30),10,11,1,.47,.7
220 CIRCLE(120,30),15,11,.7,.46,.7
230 PAINT(108,30),11
240 PAINT(93,30),11
250 CIRCLE(116,120),40,6,1.8,.45,.61
260 CIRCLE(86,114),40,6,1.7,.9,.07
270 CIRCLE(73,145),5,6,1,.9,.2
280 CIRCLE(127,142),5,6,1,.3,.55
290 CIRCLE(130,152),10,6,.6,.7,.35
300 CIRCLE(73,154),10,6,.6,.2,.8
310 LINE (77,159)-(92,159),6
320 LINE (125,157)-(108,157),6
330 LINE (92,159)-(100,130),6
340 LINE (108,157)-(100,130),6
350 CIRCLE(110,120),4,1,1,0,1,BF
360 CIRCLE(106,120),2,1,1.5,0,1,BF
370 CIRCLE(109,117),4,1,2,0,1,BF
380 CIRCLE(95,100),3,1,2,0,1,BF
390 CIRCLE(93,100),5,1,.9,0,1,BF
400 CIRCLE(94,103),3,1,.9,0,1,BF
410 CIRCLE(97,104),4,1,1,0,1,BF
420 CIRCLE(85,130),5,1,1,0,1,BF
430 CIRCLE(87,128),4,1,2,0,1,BF
440 CIRCLE(120,90),4,1,2,0,1,BF
450 CIRCLE(137,115),15,6,1.5,.6,0
460 CIRCLE(151,118),3,6,2,0,1,BF
470 GOTO 470
```

Here are some interesting statistical programs by Dr A. D. Campbell.

The first program calculates factorials, (n!). Factorial 7, or 7! is 7 × 6 × 5 × 4 × 3 × 2 × 1 = 5040. The SEGA memory limits you to 69! The John Sands program simply scrolled through the whole lot from 0! to 69! My modification gives you the choice; the lot, or an entered single value.

The second program calculates simple functions like means, standard deviations and standard errors.

The third program calculates standardised scores. This would allow teachers and others to scale marks. You specify the mean and standard deviation you would like, and the program does the scaling.

FACTORIAL

```
10 CLS
20 PRINT"                    FACTORIAL":PRINT:PRINT:PRINT:PRINT
30 POKE&HAF6E,1
40 PRINT"THIS PROGRAM CALCULATES THE VALUE OF"
50 PRINT
60 PRINT"N! FOR ALL INTEGER VALUES FROM 0! TO":PRINT
70 PRINT"69! WHERE THE MAXIMUM ALLOWABLE VALUE":PRINT
80 PRINT"OF THE SEGA PROCESSOR IS REACHED.":PRINT:PRINT:PRINT
90 PRINT"SWITCH PRINTER ON":PRINT:PRINT:PRINT
100 PRINT"DO YOU WANT A SINGLE VALUE OR A FULL":PRINT
110 PRINT"LIST?    ENTER [SV] OR [FL].":PRINT
120 INPUT J$
130 IF J$="SV" THEN 310
140 IF J$="FL" THEN 160
150 IF J$<>"SV"ORJ$<>"FL"THEN120
160 N=69
170 LPRINT " N"," N!":LPRINT
180 LPRINT " 0!"," 1"
190 LPRINT " 1!"," 1"
200 FOR M=2 TO N STEP 1
210 L=M
220 F1=M
230 FOR I=1 TO M-1 STEP 1
240 L=L-1
250 F1=F1*L
260 NEXT I
270 LPRINT M;"!",F1
280 NEXT M
290 PRINT:PRINT:PRINT"CALCULATIONS COMPLETED"
300 END
310 PRINT:INPUT"ENTER VALUE OF N:   N = ";M
320 L=M
330 F1=M
340 FORI=1TOM-1STEP1
350 L=L-1
360 F1=F1*L
370 NEXTI
380 LPRINTM;"!", F1
390 PRINT:PRINT:PRINT"CALCULATION COMPLETED"
400 END
```

STANDARDISED SCORES

```
2 CLS
5 PRINT"           STANDARDISED SCORES":PRINT:PRINT
50 POKE&HAF6E,1
100 REM PROGRAM SS
110 DIM X(200),Y(200),Z(200)
120 PRINT"THIS PROGRAM CALCULATES TWO SETS OF":PRINT
130 PRINT"STANDARDISED SCORES:":PRINT:PRINT
140 PRINT"Y(I):   MEAN = 0":PRINT
145 PRINT"        STANDARD DEVIATION = 1":PRINT:PRINT
150 PRINT"Z(I):   MEAN = Mu":PRINT
```

```
155 PRINT"        STANDARD DEVIATION = Sigma":PRINT:PRINT
160 PRINT"SWITCH PRINTER ON":PRINT:PRINT
170 PRINT"ENTER DESIRED MEAN VALUE OF Z(I) ARRAY"
175 PRINT"Mu = ";:INPUTM2:PRINT
190 PRINT:PRINT"ENTER DESIRED STANDARD DEVIATION OF":PRINT
195 PRINT"Z(I) ARRAY, Sigma = ";:INPUTS2
210 PRINT:PRINT:INPUT"ENTER NUMBER OF DATA VALUES:  ";N
220 PRINT:PRINT:PRINT"NOW ENTER SINGLY THE";N;" DATA VALUES":PRINT
230 FOR I=1 TO N
240 INPUT X(I)
250 NEXT I
260 REM CALCULATE MEAN
270 Y1=0
280 FOR I=1 TO N STEP 1
290 Y1=Y1+X(I)/N
300 NEXT I
310 REM CALCULATE STANDARD DEVIATION
320 Y2=0
330 FOR I=1 TO N STEP 1
340 Y2=Y2+((X(I)-Y1)*(X(I)-Y1)/(N-1))
350 NEXT I
360 S1=Y2^.5
370 REM CALCULATE Y(I) AND Z(I)
375 PRINT
380 LPRINT:LPRINTCHR$(27);"E";"X(I)","      Y(I)":LPRINT:LPRINTCHR$(27);"F";
390 FOR I=1 TO N STEP 1
400 Y(I)=(X(I)-Y1)/S1
410 Z(I)=M2+S2*Y(I)
420 LPRINT X(I),Y(I)
430 NEXT I
435 PRINT:LPRINT
440 LPRINT:LPRINT:LPRINTCHR$(27);"E";"X(I)","      Z(I)":LPRINT:LPRINTCHR$(27);"F
";
450 FOR I=1 TO N STEP 1
460 LPRINT X(I),Z(I)
470 NEXT I
480 PRINT"CALCULATIONS COMPLETED"
490 END
```

SINGLE VARIABLE STATISTICS

```
20 CLS
40 POKE&HAF6E,1
50 PRINT"       SINGLE VARIABLE STATISTICS":PRINT:PRINT:PRINT:PRINT:PRINT
60 PRINT:PRINT"SWITCH PRINTER ON":PRINT:PRINT
100 REM PROGRAM SVS
110 DIM X(200)
120 PRINT:PRINT"THIS PROGRAM CALCULATES THE MOST":PRINT
130 PRINT"COMMON SINGLE VARIABLE X STATISTICS OF"
140 PRINT"A SET OF NUMBERS.":PRINT
155 PRINT:PRINT:PRINT
160 INPUT "ENTER TOTAL NUMBER OF DATA VALUES: ";N
165 PRINT:PRINT:PRINT:PRINT
170 PRINT"ENTER NOW, ONE AT A TIME, THE":PRINT
180 PRINT"ACTUAL VALUES:":PRINT
190 FOR I=1 TO N STEP 1
200 INPUT X(I)
201 IF X(I)=0 THEN PRINT"CAREFUL, THERE IS A ZERO VALUE IN DATA":X(I)=1E-98
202 X(I)=X(I)+1E-98
210 NEXT I
220 Y1=0
230 M=1
240 FOR I=1 TO N STEP 1
250 Y1=Y1+X(I)
260 NEXT I
270 Y1=Y1/N
280 Y3=0
290 Y4=0
300 Y2=0
310 Q=0
```

```
320 H=0
330 G=1
340 FOR I=1 TO N STEP 1
350 Q=Q+X(I)*X(I)
360 H=H+(1/X(I))
370 G=G*X(I)
375 Z=(X(I)-Y1)
380 Y2=Y2+Z*Z
390 Y3=Y3+Z*Z*Z
400 Y4=Y4+Z*Z*Z*Z
410 NEXT I
420 Q=(Q/N)^.5
430 H=N/H
438 IF G<=0 THENPRINT:PRINT"GEOMETRIC MEAN CAN NOT BE CALCULATED!!":G=0:GOTO 450
440 G=G^(1/N)
450 Y2=Y2/N
460 Y3=Y3/N
470 Y4=Y4/N
480 C5=Y3/(Y2^1.5)
490 C8=Y4/(Y2^2)
500 V1=Y2*N/(N-1)
510 LPRINT"MEAN (1ST MOMENT ABOUT THE ORIGIN)   ",Y1
520 LPRINT"2ND MOMENT (UNBIASED) ABOUT THE MEAN",V1
530 LPRINT"2ND MOMENT ABOUT THE MEAN             ",Y2
540 LPRINT"HARMONIC MEAN                         ",H
550 LPRINT"QUADRATIC MEAN                        ",Q
560 LPRINT"GEOMETRIC MEAN                        ",G
565 IF G>0 THEN 580
568 PRINT
570 PRINT"NOTICE THAT GEOMETRIC MEAN MAY BE      PRINTED AS ZERO IF THERE ARE NEG
ATIVE OR ZERO VALUES IN THE DATA SET."
575 PRINT
580 S1=V1^0.5
590 LPRINT"STANDARD DEVIATION                    ",S1
600 LPRINT"3RD MOMENT (MEAN)                     ",Y3
610 LPRINT"4TH MOMENT (MEAN)                     ",Y4
620 LPRINT"SKEWNESS                              ",C5
630 LPRINT"KURTOSIS                              ",C8
640 S2=S1/(N^0.5)
650 LPRINT"STANDARD ERROR                        ",S2
654 PRINT:PRINT
660 PRINT"CALCULATIONS COMPLETED"
670 END
```


LIGHTNING BOLT

BASIC EXTENSION
By Michael Hadrup

Includes Load/Save/Verify Routines and
dissassembler, machine code monitor, etc. . .

AVAILABLE SOON
FOR 16k/32k/disk

Programs by B. E. Pycroft
Wainuiomata

ZERO & "0" & PRINTER

```
1 REM Patterns ZERO & "O" & Ptr '2':::by  B.E. PYCROFT.  Wainuiomata. 1985.
2 PRINT CHR$(26):PATTERNC#79,"7088888888887000":PATTERNC#48,"708898A8C8887000"
3 POKE&H8650,&H5B:POKE&H8651,&HAB:POKE&H8652,&H5D:POKE&H8653,&HA3
4 REM '[' = 'CONSOLE'. ']'= 'CURSOR'.
5 POKE&H8586,&H5C:POKE&H8587,&H87:POKE&H8588,&H5F:POKE&H8589,&HB5
6 REM '\' = 'CONT'. .  '_'= 'HCOPY'..
7 POKE&H5E62,&H48:POKE&H5E63,69:POKE&H5E64,88:POKE&H5E65,36
8 POKE&H5E36,&H50:POKE&H5E37,69:POKE&H5E38,69:POKE&H5E39,75
9 POKE&H5E0E,&H50:POKE&H5E0F,79:POKE&H5E10,75:POKE&H5E11,69
10 REM 'RAD(' = 'HEX$'. . 'TAN(' = 'PEEK'.. 'COS(' = 'POKE'..
11 POKE&H5EF2,&H43:POKE&H5EF3,65:POKE&H5EF4,76:POKE&H5EF5,76
12 REM 'SIN(' = 'CALL'.
13 POKE&H5E92,&H26:POKE&H5E93,72
14 REM 'PI' = '&H'.
15 POKE&H8654,&H40:POKE&H8655,&H8D
16 REM '@' = 'FILES':: Makes directory listing more friendly!!                ('@'
 is permanent on my basic.B.E.P.)
17 REM << Note that now the 'MOTOR' command has been DELETED in favour of '['>CO
NSOLE;; ']'>CURSOR;; '@'>FILES which occupies the same space at &H8650 to &H8655
.
18 REM > Obviously, one may alter or rearrange this program for personal taste o
r even have several Modified BASIC Disks oriented for specific tasks.
20 REM >E.N.D.<
```

DISK FILES PRINTER

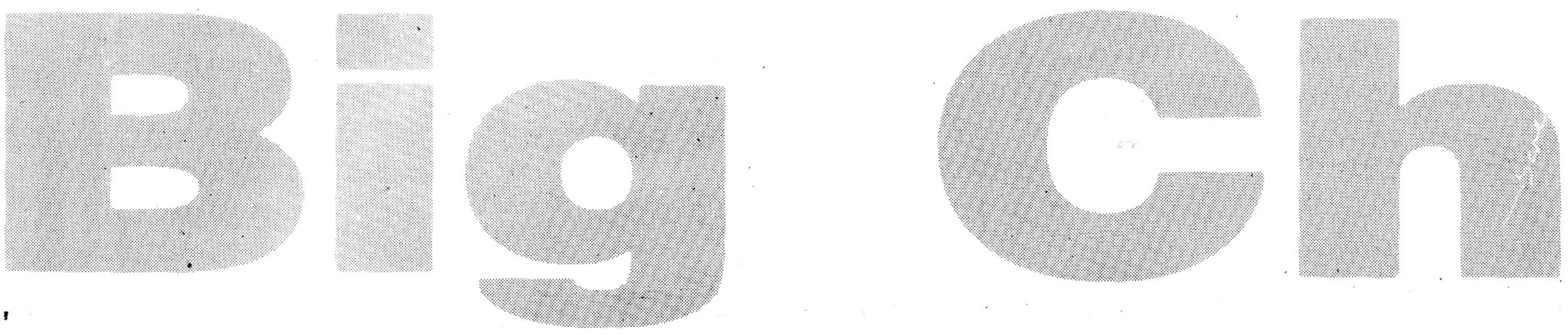
```
10 REM Disk files printer.. B. E. PYCROFT.. Wainuiomata.  1986.
12 REM You may alter the 'TAB(T*xx)' constant for other formats.
20 GOTO 300
100 TR=20:SC=1:N=1
110 DSKI$TR,SC;A$(N),0,127;A$(N+1),128,128
120 IF ASC(A$(N))>32 THEN N=N+2:SC=SC+1:IF SC<11 THEN 110
140 C=1:T=0
150 S=1:O=12
160 P$=MID$(A$(C),S,O):IF ASC(P$)< 32THEN 180
170 LPRINT TAB(T*15);P$;:T=T+1:IF T>2THENT=0:LPRINT
180 S=S+16:IF S<127 THEN 160
190 C=C+1:IF C<N THEN 150
200 LPRINT DSKF;"K bytes free.":LPRINT
210 INPUT "Another side to do? {Y/N}";I$:IF I$<>"Y" THEN PRINT "FINISHED.":GOTO
350
220 PRINT "Change disk &/or side. ":GOTO 310
300 PRINT"          Disk FILES Printer.
302 PRINT :PRINT " Three columns of program titles are  printed to fit into the
width of disk or cassette cases.
304 PRINT:PRINT "Insert a disk in drive.":PRINT
310 INPUT"Enter disk  number.>";D$
320 INPUT"Enter disk   side .>";S$
330 PRINT "<< Disk ";D$;" >>  << Side ";S$;" >>"
335 LPRINT "<< Disk ";D$;" >>  << Side ";S$;" >>"
340 GOTO 100
350 END
```

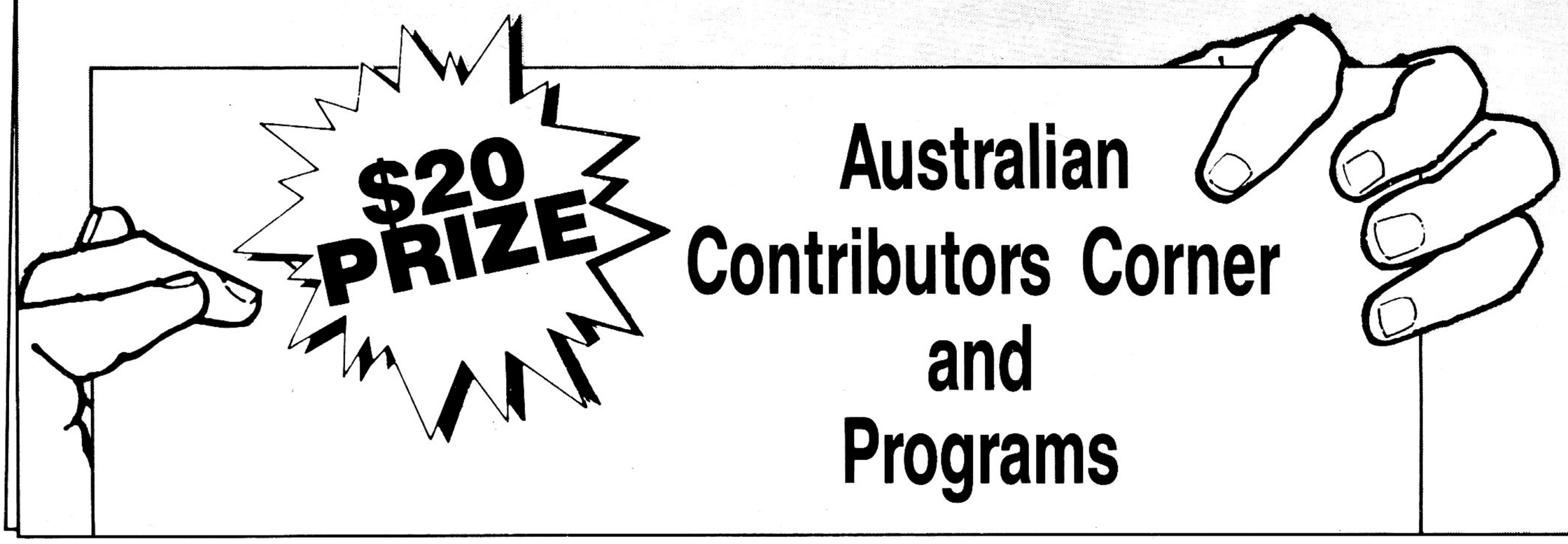
Big Characters

By Rod Cuckow

This program by Rod Cuckow draws much enlarged characters on the screen and would be great for titles, etc., in software. The characters are so large that only nine characters fit across the screen! As you can see from the program, it takes the letters from a string and converts them into large characters.

```
10 REM BIG CHR$
20 REM By Rod Cuckow 1986
30 REM ------------------
40 REM This prints very large on the
50 REM screen.
60 REM
70 REM
80 CLS:PRInT "TO RUN ON MEMORY CARTRIDGE INPUT C":PRInT :PRInT "TO RUN ON DISK I
NPUT D"
90 PRInT :PRInT "IF YOU PRESS C WHEN RUNNING ON DISK   THE PROGRAM WILL CRASH AF
TER A WHILE":PRInT
100 PRInT "OR PRESS D WHEN RUNNING ON CARTRIDGE  NOTHING"
110 CURSOR0,20:PRInT "THIS INTRO FOR YOUR INFORMATION ONLY  PRINT OUT ALREADY SE
NT IS PROGRAM    PROPER"
120 CURSOR 0,10:INPUT X$
130 IF X$="C"ORX$="c" THEN 170
140 IF X$="D"ORX$="d" THEN 180
150 GOTO 80
160 DATA 128,64,32,16,8,4,2,1
170 DEF FN C(X)=&H10C0+((X-32)*8):GOTO 190
180 DEF FN C(X)=&HA316+(X*8)
190 Y1=17:X1=0:CLS
200 REM
210 REM This is the input string
220 IP$=" BIG CHR$ ~PROGRAM~ ========= *********"
230 REM
240 REM This prints it large
245 REM ----------------------------
250 SCREEN 2,2:CLS
260 FOR I=1 TO LEN(IP$):A$=MID$(IP$,I,1)
270 X=ASC(A$):X=FN C(X):Z=X1
280 FOR A=X TO X+7
290 Z=Z+4:P=PEEK(A):RESTORE
300 FOR B=0 TO 7
310 READ D
320 IF Z>191 THEN SCREEN 1,1:CLS:PRInT "END":END
330 IF D AND P THEN CURSOR Y1+B*4,Z:PRInT CHR$(236)
340 REM This adds sound to it all!
350 REM IF P<>0 THEN SOUND 1,3500,15:SOUND 0
360 NEXT B,A:Y1=Y1+24:IF Y1>240 THEN Y1=17:X1=X1+34
370 COLORINT(RND(1)*13)+1
380 NEXT
390 REM
400 REM This Ends The Routine
410 REM ----------------------------
420 COLOR1
430 CURSOR100,160:PRInT "MODIFIED for Disk"
440 CURSOR 100,169:PRInT "by         or Cartridge"
450 CURSOR 100,182:PRInT CHR$(17);"Rod Cuckow":PRInT CHR$(16)
460 GOTO 460
```





Australian Contributors Corner and Programs

Martin Schult of South Australia has contributed this program for our New Zealand magazine, so now you can create your own D&D character fast.

Dungeons & Dragons Character Generator

HERE'S ONE FOR ALL YOU DUNGEON AND DRAGONS PLAYERS. THIS PROGRAM WILL GENERATE ALL 10 CLASSES OF CHARACTERS UP TO LEVEL 5.
IT FOLLOWS THE GUIDES AS SET OUT IN THE OFFICIAL D&D BOOKS AS CLOSELY AS SPACE WOULD PERMIT.
YOU REQUIRE A 3B CARTRIDGE FOR THIS PROGRAM AND AFTER RUNNING IT, THERE ARE ONLY AROUND 550 BYTES LEFT FREE.

```
10 REM D&D CHARACTERS
20 RESTORE
30 SCREEN2,2:CLS:COLOR1,13,(0,0)-(255,191),13
40 DIMH$(45):FORH=0TO44:READH$(H)
50 IFH=18THENX=55:X1=45
60 IFH=27THENX=0:X1=90
70 CURSOR25+X,20+X1:PRINTCHR$(17);H$(H);CHR$(16):SOUND1,3000-SD,15
80 SOUND0:SD=SD+50
90 X=X+12:NEXTH
100 DATA D,U,N,G,E,O,N,S, ,&, ,D,R,A,G,O,N,S,C,H,A,R,A,C,T,E,R,G,E,N,E,R,A,T,I,O,N, ,P,R,O,G,R,A,M,
110 LINE(52,135)-(218,135),1:BEEP
120 LINE-(218,152),1:BEEP
130 LINE-(52,152),1:BEEP
140 LINE-(52,135),1:BEEP
150 CURSOR57,140:PRINT"BY...Martin Schult  SEP'86
160 DIMWE$(40):DIMPO$(75):DIMFA$(12):DIMFB$(12):DIMFC$(11)
170 DIMFD$(10):DIMFE$(12):DIMFF$(12)
180 DIMFG$(20):DIMFH$(20):DIMFI$(20)
190 DIMFJ$(11):DIMFK$(12):DIMFL$(10)
200 FORW=1TO40:READWE$(W):NEXTW
210 FORP=1TO75:READPO$(P):NEXTP
220 FORFA=1TO12:READFA$(FA):NEXTFA
230 FORFB=1TO12:READFB$(FB):NEXTFB
240 FORFC=1TO11:READFC$(FC):NEXTFC
250 FORFD=1TO10:READFD$(FD):NEXTFD
260 FORFE=1TO12:READFE$(FE):NEXTFE
270 FORFF=1TO12:READFF$(FF):NEXTFF
280 FORFG=1TO20:READFG$(FG):NEXTFG
290 FORFH=1TO20:READFH$(FH):NEXTFH
300 FORFI=1TO20:READFI$(FI):NEXTFI
310 FORFJ=1TO11:READFJ$(FJ):NEXTFJ
320 FORFK=1TO12:READFK$(FK):NEXTFK
330 FORFL=1TO10:READFL$(FL):NEXTFL
340 PR$="Press spacebar to continue
350 MI$=".....Misc Info.....
.........
360 MC$=".....Misc Info (cont)..
.........
370 EA$=".....WEAPONS POSSESSED.
.........
380 SK$=".....SPELLS KNOWN.....
.........
390 LL$="1st Level    2nd Level    3rd Level
```

CHARACTER SHEET

NAME: _____ RACE: _____ CLASS: _____ LEVEL: _____

ABILITY SCORES:

Strength: ..... _____

Intelligence: ..... _____

Wisdom: ..... _____

Dexterity: ..... _____

Constitution: ..... _____

Charisma: ..... _____

Armour Class: ..... _____

Armour: ..... _____

Damage Adjust: ..... _____

Reac/Att Adj: ..... _____

Defensive Adj: ..... _____

Magical Att Adj: ..... _____

Open Doors: ..... _____

Speech: ..... _____

HIT POINTS:

_____ (incl. asjust's)

SAVING THROWS

Death Ray or Poison	Magic Wands	Paralysis or Turn To Stone	Dragon Breath	Rods Staves or Spell

vs UNDEAD

Skel	Zom	Ghou	Wight	Mum	Spec	Vamp

THIEVES ABILITIES

Pick Pocket	Open Lock	Find & Remove Trap	Move Silent	Hide In Shadow	Hear Noise

WEAPONS

POSSESSIONS

MAGICAL ITEMS

MAGICAL SPELLS (inc. Illusion)

1st Level:

2nd Level:

3rd Level:

CLERICAL SPELLS (inc. Druid)

1st Level:

2nd Level:

3rd Level:

```
400 Z7$="  11    -    --    --    --    -
--    --
410 Z8$="   9   11    --    --    --    -
--    --
420 Z1$="   7    9    11    --    --    -
--    --
430 Z2$="   T    7    9    11    --    -
--    --
440 Z3$="   T    T    7    9    11    -
--    --
450 Z4$="   D    T    T    7    9    11
--    --
460 Z5$="   D    D    T    T    7    9
11    --
470 T1$="   30Z    25Z    20Z    15Z    10Z
    10Z
480 T2$="   35Z    29Z    25Z    21Z    15Z
    10Z
490 T3$="   40Z    33Z    30Z    27Z    20Z
    15Z
500 T4$="   45Z    37Z    35Z    33Z    25Z
    15Z
510 T5$="   50Z    42Z    40Z    40Z    31Z
    20Z
520 GOTO730
530 M1=INT(RND(-1)*6)+1
540 M2=INT(RND(-1)*6)+1
550 M3=INT(RND(-1)*6)+1
560 MT=M1+M2+M3
570 GD=(MT*10):RETURN
580 AA=INT(RND(-1)*11)
590 IFAA<5THEN580
600 AC=(AA+DF)
610 IFAC=10THENAC$="NO ARMOUR...(HA HA
HAHA)
620 IFAC=9THENAC$="SHIELD ONLY
630 IFAC=8THENAC$="LEATHER ARMOUR
640 IFAC=7THENAC$="STUDDED LEATHER & S
HIELD
650 IFAC=6THENAC$="RING MAIL & SHIELD
660 IFAC=5THENAC$="SCALE MAIL & SHIELD

670 IFAC=4THENAC$="CHAIN MAIL & SHIELD

680 IFAC=3THENAC$="BANDED MAIL
690 IFAC=2THENAC$="PLATE MAIL
700 IFAC=1THENAC$="PLATE MAIL & SHIELD
710 IFAC=0THENAC$="PLATE MAIL, SHIELD
& HELMET
720 RETURN
730 SCREEN1,1:CLS:COLOR1,4
740 GOSUB530:CURSOR12,12:PRINT"Please
wait...
750 L=L+1
760 Z=INT(RND(-1)*6)+1
770 Z2=INT(RND(-1)*6)+1
780 Z3=INT(RND(-1)*6)+1
790 ZZ=Z+Z2+Z3
800 IFL=1THEN860
810 IFL=2THEN990
820 IFL=3THEN1070
830 IFL=4THEN1140
840 IFL=5THEN1240
850 IFL=6THEN1370
860 ST=ZZ
870 IFST<13THEN760
880 IFST>11ANDST<16THENDA=1:OD$="1-2
890 IFST>15ANDST<18THENDA=2:SP=1:OD$="
1-3
900 IFST=18THENGOTO920
910 GOTO750
920 SS=INT(RND(-1)*100)+1
930 IFSS>0ANDSS<51THENDA=3:SP=2:OD$="1
-3
940 IFSS>50ANDSS<76THENDA=4:SP=2:OD$="
1-4
950 IFSS>75ANDSS<91THENDA=4:SP=3:OD$="
1-4
960 IFSS>90ANDSS<100THENDA=5:SP=3:OD$=
"1-4(1)
970 IFSS=100THENDA=6:SP=3:OD$="1-5(2)
980 GOTO750
990 CO=ZZ
1000 A=ST/CO
1010 IFA<0.67ORA>1.5THEN760
1020 IFCO=15THENCP=1
1030 IFCO=16THENCP=2
1040 IFCO=17THENCP=3
1050 IFCO=18THENCP=4
1060 GOTO750
1070 IN=ZZ
1080 IFIN=3THENI$="* Difficult speech
- Illiterate
1090 IFIN>3ANDIN<6THENI$="* Easy speec
h but illiterate
1100 IFIN>5ANDIN<9THENI$="* Speech bar
ely literate
1110 IFIN>8ANDIN<13THENI$="* Literate
in native tongue
1120 IFIN>12THENI$="* Fluent speech
1130 GOTO750
1140 DX=ZZ
1150 IFDX=3THENDF=4:RA=-3
1160 IFDX>3ANDDX<6THENDF=3:RA=-2
1170 IFDX>5ANDDX<9THENDF=2:RA=-1
1180 IFDX>8ANDDX<15THENDF=0:RA=0
1190 IFDX=15THENDF=-1:RA=0
1200 IFDX=16THENDF=-2:RA=1
1210 IFDX=17THENDF=-3:RA=2
1220 IFDX=18THENDF=-4:RA=3
1230 GOTO750
1240 WI=ZZ
1250 B=IN/WI
1260 IFB<0.67ORB>1.5THEN760
1270 IFWI<3THENMA=-5
1280 IFWI=3THENMA=-3
1290 IFWI=4THENMA=-2
1300 IFWI>4ANDWI<8THENMA=-1
1310 IFWI>7ANDWI<15THENMA=0
1320 IFWI=15THENMA=1
```

```
1330 IFWI=16THENMA=2
1340 IFWI=17THENMA=3
1350 IFWI=18THENMA=4
1360 GOTO750
1370 CH=ZZ
1380 L=0
1390 IF(ST+CO+IN+DX+WI+CH)<61THEN740
1400 GOSUB5220
1410 CLS
1420 C1$="":C2$="":C3$="":C4$="":C5$="
":C6$="":C7$="":C8$="":C9$="":C0$=""
1430 IFST>11ANDCO>7THENC1$="(F)IGHTER
"
1440 IFST>11ANDWI>12THENC2$="(P)ALADIN
 "
1450 IFST>12ANDIN>12ANDCO>13THENC3$="(
R)ANGER "
1460 IFDX>12ANDIN>10THENC4$="(T)HIEF "

1470 IFST>12ANDDX>12THENC5$="(A)SSASSI
N "
1480 IFST>14ANDWI>14ANDDX>14THENC6$="M
(O)NK "
1490 IFWI>11THENC7$="(C)LERIC "
1500 IFWI>11ANDCH>14THENC8$="(D)RUID "

1510 IFIN>12THENC9$="(M)AGIC-USER "
1520 IFIN>14ANDDX>15THENC0$="(I)LLUSIO
NIST "
1530 SCREEN1,1:CLS:COLOR1,13
1540 PRINT:PRINTTAB(10);"CLASS SELECTI
ON
1550 PRINTTAB(9);"
1560 CURSOR0,5:PRINTC1$;C2$;C3$;C4$;C5
$;C6$;C7$;C8$;C9$;C0$:BEEP:BEEP
1570 CURSOR0,18:PRINT"Please select yo
ur character class
1580 C$=INKEY$
1590 IFC$=""THEN1580
1600 IFC1$=""THEN1620
1610 IFC$="F"THEN2020
1620 IFC2$=""THEN1640
1630 IFC$="P"THEN2200
1640 IFC3$=""THEN1660
1650 IFC$="R"THEN2430
1660 IFC4$=""THEN1680
1670 IFC$="T"THEN2600
1680 IFC5$=""THEN1700
1690 IFC$="A"THEN2900
1700 IFC6$=""THEN1720
1710 IFC$="O"THEN3130
1720 IFC7$=""THEN1740
1730 IFC$="C"THEN3490
1740 IFC8$=""THEN1760
1750 IFC$="D"THEN3820
1760 IFC9$=""THEN1780
1770 IFC$="M"THEN4090
1780 IFC0$=""THEN1810
1790 IFC$="I"THEN4340
1800 GOTO1820
1810 GOTO1530
1820 CLS
1830 PRINT:PRINT:PRINT
1840 PRINT "Character level range.....
1 to 5":PRINT
1850 PRINT "At which level is Characte
r to
1860 INPUT"start? ";LL
1870 IFLL>5THENPRINT"Too high BOZO!! T
ry again ":GOTO1840
1880 Y=0
1890 Y=Y+1
1900 HP=INT(RND(-1)*HD)+1
1910 HP(Y)=HP
1920 IFY<>LLTHEN1890
1930 FORY=0TOLL-1
1940 HP=HP+HP(Y)
1950 NEXTY
1960 HT=HP+SP+CP
1970 IFHT<=0THEN1990
1980 GOTO2000
1990 HT=LL
2000 Y=0
2010 RETURN
2020 REM FIGHTER SKILLS
2030 CLS
2040 CN$="FIGHTER":HD=10
2050 GOSUB1820
2060 GOSUB580
2070 IFLL<3THENST$="..14....16.......15
.......17......17..
2080 IFLL=3ORLL=4THENST$="..13....15..
....14.......16......16..
2090 IFLL=5THENST$="..11....13......12
.......13......14..
2100 IFLL<4THENWE=4
2110 IFLL>3THENWE=5
2120 CLS:PRINTTAB(7);CN$;" :-"
2130 PRINT:PRINTMI$
2140 PRINT:PRINT"Please refer to page
22 of the PlayersHandbook for a detail
ed description ofyour abilities.
2150 CURSOR5,22:PRINTPR$
2160 IN$=INKEY$:IFIN$=""THEN2160
2170 GOSUB5050
2180 GOSUB4590
2190 GOTO5380
2200 REM PALADIN SKILLS
2210 CN$="PALADIN":HD=10
2220 GOSUB1820
2230 IFLL=1THENZ6$=Z7$
2240 IFLL=2THENZ6$=Z8$
2250 IFLL=3THENZ6$=Z1$
2260 IFLL=4THENZ6$=Z2$
2270 IFLL=5THENZ6$=Z3$
2280 IFLL<3THENST$="..14....16.......15
.......17......17..
2290 IFLL=3ORLL=4THENST$="..13....15..
....14.......16......16..
2300 IFLL=5THENST$="..11....13......12
```

```
.....13.....14..
2310 GOSUB580
2320 IFLL<4THENWE=3
2330 IFLL>3THENWE=4
2340 CLS:PRINTTAB(7);CN$;" :-"
2350 PRINT:PRINTMI$
2360 PRINT:PRINT"Please refer to pages
 22 & 24 of the  Players Handbook for
a detailed des-  cription of your abil
ities.
2370 CURSOR5,22:PRINTPR$
2380 IN$=INKEY$:IFIN$=""THEN2380
2390 GOSUB5050
2400 GOSUB4590
2410 GOSUB4700
2420 GOTO5380
2430 REM RANGER SKILLS
2440 CN$="RANGER":HD=10
2450 GOSUB1820
2460 IFLL<3THENST$="..14....16......15
.......17......17..
2470 IFLL=3ORLL=4THENST$="..13....15..
....14.......16......16..
2480 IFLL=5THENST$="..11....13......12
.......13......14..
2490 GOSUB580
2500 IFLL<4THENWE=3
2510 IFLL>3THENWE=4
2520 CLS:PRINTTAB(7);CN$;" :-"
2530 PRINT:PRINTMI$
2540 PRINT:PRINT"Please refer to pages
 24 & 25 of the  Players Handbook for
a detailed des-  cription of your abil
ities.
2550 CURSOR5,22:PRINTPR$
2560 IN$=INKEY$:IFIN$=""THEN2560
2570 GOSUB5050
2580 GOSUB4590
2590 GOTO5380
2600 REM THIEF SKILLS
2610 CN$="THIEF":HD=6
2620 GOSUB1820
2630 IFLL<5THENST$="..13....14......12
.......16......15..
2640 IFLL=5THENST$="..12....12......11
.......15......13..
2650 IFLL=1THENT6$=T1$
2660 IFLL=2THENT6$=T2$
2670 IFLL=3THENT6$=T3$
2680 IFLL=4THENT6$=T4$
2690 IFLL=5THENT6$=T5$
2700 AC=8+DF:AC$="LEATHER ARMOUR
2710 CLS:PRINTTAB(7);CN$;" :-"
2720 PRINT:PRINTMI$
2730 PRINT:PRINT"Please refer to pages
 26-28 of the    Players Handbook for
a detailed des-  cription of your abil
ities.
2740 CURSOR5,22:PRINTPR$
2750 IN$=INKEY$:IFIN$=""THEN2750
2760 GOSUB5050
2770 GOSUB4590
2780 GOSUB4790
2790 GOSUB4900
2800 CLS:PRINTTAB(7);CN$;" :-"
2810 PRINT:PRINTEA$
2820 FORA=1TO50:NEXTA
2830 PRINT:PRINT
2840 PRINTTAB(5);"2 DAGGERS":BEEP
2850 PRINTTAB(5);"SHORT SWORD":BEEP
2860 PRINTTAB(5);"BLOW PIPE & 6 DARTS"
 :BEEP
2870 CURSOR5,22:PRINTPR$
2880 IN$=INKEY$:IFIN$=""THEN2880
2890 GOTO5490
2900 REM ASSASSIN SKILLS
2910 CN$="ASSASSIN":HD=6
2920 GOSUB1820
2930 IFLL<5THENST$="..13....14......12
.......16......15..
2940 IFLL=5THENST$="..12....12......11
.......15......13..
2950 IFLL=1THENT6$=T1$
2960 IFLL=2THENT6$=T2$
2970 IFLL=3THENT6$=T3$
2980 IFLL=4THENT6$=T4$
2990 IFLL=5THENT6$=T5$
3000 IFLL<4THENWE=3
3010 IFLL>3THENWE=4
3020 AC=8+DF:AC$="LEATHER ARMOUR
3030 CLS:PRINTTAB(7);CN$;" :-"
3040 PRINT:PRINTMI$
3050 PRINT:PRINT"Please refer to pages
 28-30 of the    Players Handbook for
a detailed des-  cription of your abil
ities.
3060 CURSOR5,22:PRINTPR$
3070 IN$=INKEY$:IFIN$=""THEN3070
3080 GOSUB5050
3090 GOSUB4590
3100 GOSUB4790
3110 GOSUB4900
3120 GOTO5380
3130 REM MONK SKILLS
3140 CN$="MONK":HD=6:DF=0:SP=0
3150 GOSUB1820
3160 IFLL<5THENST$="..13....14......12
.......16......15..
3170 IFLL=5THENST$="..12....12......11
.......15......13..
3180 IFLL=1THENZ6$=Z1$:AC=10
3190 IFLL=1THENT6$=T1$:SA$="NIL
3200 IFLL=2THENZ6$=Z2$:AC=9
3210 IFLL=2THENT6$=T2$:SA$="NIL
3220 IFLL=3THENZ6$=Z3$:AC=8
3230 IFLL=3THENT6$=T3$:SA$="A
3240 IFLL=4THENZ6$=Z4$:AC=7
3250 IFLL=4THENT6$=T4$:SA$="A & B
3260 IFLL=5THENZ6$=Z5$:AC=7
3270 IFLL=5THENT6$=T5$:SA$="A,B & C
```

```
3280 AC$="NO ARMOUR...(not even a shie
ld!!!)
3290 CLS:PRINTTAB(7);CN$;" :-"
3300 PRINT:PRINTMI$
3310 PRINT:PRINT"Please refer to pages
 30-32 of the    Players Handbook for
a detailed des-  cription of your abil
ities.
3320 PRINT:PRINT"Special Abilities are
...
   ";SA$
3330 CURSOR5,22:PRINTPR$
3340 IN$=INKEY$:IFIN$=""THEN3340
3350 GOSUB5050
3360 GOSUB4590
3370 GOSUB4700
3380 GOSUB4790
3390 CLS:PRINTTAB(7);CN$;" :-"
3400 PRINT:PRINTEA$
3410 FORA=1TO50:NEXTA
3420 PRINT:PRINT:PRINTTAB(5);"2 DAGGER
S":BEEP
3430 PRINTTAB(5);"HAND AXE":BEEP
3440 PRINTTAB(5);"STAFF":BEEP
3450 PRINTTAB(5);"SPEAR":BEEP
3460 CURSOR5,22:PRINTPR$
3470 IN$=INKEY$:IFIN$=""THEN3470
3480 GOTO5490
3490 REM CLERIC SKILLS
3500 CN$="CLERIC":HD=8
3510 GOSUB 1820
3520 GOSUB580
3530 IFLL=1THENZ6$=Z1$
3540 IFLL=2THENZ6$=Z2$
3550 IFLL=3THENZ6$=Z3$
3560 IFLL=4THENZ6$=Z4$
3570 IFLL=5THENZ6$=Z5$
3580 IFLL<4THENST$="..10....14......13
.......16......15..
3590 IFLL>3THENST$="...9....13......12
.......15......14..
3600 IFLL=1THENL1=3
3610 IFLL=2THENL1=4
3620 IFLL=3THENL1=4:L2=1
3630 IFLL=4THENL1=4:L2=3
3640 IFLL=5THENL1=5:L2=4
3650 CLS:PRINTTAB(7);CN$;" :-"
3660 PRINT:PRINTMI$
3670 PRINT:PRINT"Please refer to page
20 of the PlayersHandbook for a detail
ed description ofyour abilities.
3680 CURSOR5,22:PRINTPR$
3690 IN$=INKEY$:IFIN$=""THEN3690
3700 GOSUB5050
3710 GOSUB4590
3720 GOSUB4700
3730 GOSUB5610
3740 CLS:PRINTTAB(7);CN$;" :-"
3750 PRINT:PRINTEA$
3760 PRINT:PRINTTAB(5);"CLUB":BEEP
3770 PRINTTAB(5);"STAFF":BEEP
3780 IFLL=5THENPRINTTAB(5);"HAMMER":BE
EP
3790 CURSOR5,22:PRINTPR$
3800 IN$=INKEY$:IFIN$=""THEN3800
3810 GOTO5490'
3820 REM DRUID SKILLS
3830 CN$="DRUID":HD=8
3840 GOSUB1820
3850 IFLL<4THENST$="..10....14......13
.......16......15..
3860 IFLL>3THENST$="..09....13......12
.......15......14..
3870 AC=8+DF:IFAC>10THENAC=10
3880 AC$="LEATHER ARMOUR
3890 IFLL=1THENL1=2
3900 IFLL=2THENL1=3:L2=1
3910 IFLL=3THENL1=3:L2=2:L3=1
3920 IFLL=4THENL1=4:L2=3:L3=2
3930 IFLL=5THENL1=5:L2=4:L3=3
3940 CLS:PRINTTAB(7);CN$;" :-"
3950 PRINT:PRINTMI$
3960 PRINT:PRINT"Please refer to pages
 20 & 21 of the  Players Handbook for
a detailed des-  cription of your abil
ities.
3970 CURSOR5,22:PRINTPR$
3980 IN$=INKEY$:IFIN$=""THEN3980
3990 GOSUB5050
4000 GOSUB4590
4010 GOSUB5880
4020 CLS:PRINTTAB(7);CN$;" :-"
4030 PRINT:PRINTEA$
4040 PRINT:PRINTTAB(5);"2 DAGGERS
4050 PRINTTAB(5);"STAFF
4060 CURSOR5,22:PRINTPR$
4070 IN$=INKEY$:IFIN$=""THEN4070
4080 GOTO5490
4090 REM MAGIC-USER SKILLS
4100 CN$="MAGIC-USER":HD=4
4110 GOSUB1820
4120 AC=10+DF:AC$="NO ARMOUR...(HA HA!
!!!)
4130 ST$="..14....11......13.......15.
.....12..
4140 IFLL=1THENL1=4
4150 IFLL=2THENL1=4:L2=2
4160 IFLL=3THENL1=5:L2=3:L3=1
4170 IFLL=4THENL1=5:L2=4:L3=2
4180 IFLL=5THENL1=6:L2=5:L3=4
4190 CLS:PRINTTAB(7);CN$;":- "
4200 PRINT:PRINTMI$
4210 PRINT:PRINT"Please refer to pages
 25 & 26 of the  Players Handbook for
a detailed des-  cription of your abil
ities.
4220 CURSOR5,22:PRINTPR$
4230 IN$=INKEY$:IFIN$=""THEN4230
4240 GOSUB5050
4250 GOSUB4590
```

```
4260 GOSUB6150
4270 CLS:PRINTTAB(7);CN$;" :-"
4280 PRINT:PRINTEA$
4290 PRINT:PRINTTAB(5);"2 DAGGERS
4300 PRINTTAB(5);"STAFF
4310 CURSOR5,22:PRINTPR$
4320 IN$=INKEY$:IFIN$=""THEN4320
4330 GOTO5490
4340 REM ILLUSIONIST SKILLS
4350 CN$="ILLUSIONIST":HD=4
4360 GOSUB1820
4370 AC=10+DF:AC$="NO ARMOUR...(HA HA!
!!!)
4380 SI$="..14....11......13.......15.
.....12..
4390 IFLL=1THENL1=3
4400 IFLL=2THENL1=4
4410 IFLL=3THENL1=4:L2=2
4420 IFLL=4THENL1=5:L2=3
4430 IFLL=5THENL1=6:L2=4:L3=2
4440 CLS:PRINTTAB(7);CN$;" :-"
4450 PRINT:PRINTMI$
4460 PRINT:PRINT"Please refer to page
26 of the PlayersHandbook for a detail
ed description ofyour abilities.
4470 CURSOR5,22:PRINTPR$
4480 IN$=INKEY$:IFIN$=""THEN4480
4490 GOSUB5050
4500 GOSUB4590
4510 GOSUB6420
4520 CLS:PRINTTAB(7);CN$;" :-"
4530 PRINT:PRINTEA$
4540 PRINT:PRINTTAB(5);"2 DAGGERS
4550 PRINTTAB(5);"STAFF
4560 CURSOR5,22:PRINTPR$
4570 IN$=INKEY$:IFIN$=""THEN4570
4580 GOTO5490
4590 CLS:PRINTTAB(7);CN$;" :-"
4600 PRINT:PRINT"          SAVING THRO
W TABLE
4610 PRINT:PRINT"DEATH |     |PARALYSI
S|      | RODS
4620 PRINT"RAY OR|MAGIC| OR TURN |DRAG
ON|STAVES
4630 PRINT"POISON|WANDS|TO STONE |BREA
TH|OR SPELL
4640 PRINT"[redacted]
4650 PRINT:PRINTST$
4660 CURSOR5,16:PRINT"For any other fo
rms of attack,          your DM will inf
orm you of the          required Saving
Throw.
4670 CURSOR5,22:PRINTPR$
4680 IN$=INKEY$:IFIN$=""THEN4680
4690 RETURN
4700 CLS:PRINTTAB(7);CN$;" :-"
4710 PRINT:PRINT"     PALADIN/CLERIC/M
ONK vs UNDEAD
4720 PRINT:PRINT"SKEL ZOM GHOU WIGT WR
AI MUM SPEC VAMP
4730 PRINT"[redacted]
4740 PRINT:PRINTZ6$
4750 CURSOR5,16:PRINT"For any other at
tack from an          'Undead', your D
M will inform         you of the requi
red score.
4760 CURSOR5,22:PRINTPR$
4770 IN$=INKEY$:IFIN$=""THEN4770
4780 RETURN
4790 CLS:PRINTTAB(7);CN$;" :-"
4800 PRINT:PRINT"   THEIVES/ASSASSINS/
MONK ABILITIES
4810 PRINT:PRINT"          |      |FIND&|
  |HIDE |
4820 PRINT" PICK |OPEN |REMOV|MOVE | I
N  | HEAR
4830 PRINT"POCKET|LOCKS|TRAPS|SILNT|SH
ADO| NOISE
4840 PRINT"[redacted]
4850 PRINT:PRINTT6$
4860 CURSOR5,16:PRINT"For your chance
of success on          other abilities
you may have,          please consult y
our DM.
4870 CURSOR5,22:PRINTPR$
4880 IN$=INKEY$:IFIN$=""THEN4880
4890 RETURN
4900 CLS:PRINTTAB(7);CN$;" :-"
4910 PRINT:PRINTMC$
4920 PRINT"Depending on the race which
 you chose,you have the following adju
stments to your special abilities....
4930 PRINT:PRINT"          |      |FIND&|
   |HIDE |
4940 PRINT" PICK |OPEN |REMOV|MOVE | I
N  | HEAR
4950 PRINT"POCKET|LOCKS|TRAPS|SILNT|SH
ADO| NOISE
4960 PRINT"[redacted]
4970 PRINT"D  -     +10%  +15%   -
-     -
4980 PRINT:PRINT"E +5%   -05%   --    +
5%   +10%  +05%
4990 PRINT:PRINT"G  -     +05%  +10%  +
5%   +05%  +10%
5000 PRINT:PRINT"D=Dwarf  E=Elf  G=Gno
me
5010 PRINT"No adj for Humans.
5020 CURSOR5,22:PRINTPR$
5030 IN$=INKEY$:IFIN$=""THEN5030
5040 RETURN
5050 REM CHARACTER INFO
5060 CLS:PRINTTAB(7);CN$;" :-"
5070 PRINT:PRINT"Starting Level.......
";LL
5080 PRINT"Armour Class.........";AC;"
 + AC already
```

```
5090 PRINT"Defensive Adjust....."
;DF;" adjusted
5100 PRINT"Hit Points....."
;HT
5110 PRINT"Damage Adjust....."
;DA
5120 PRINT"Reaction/Attack Adj.."
;RA
5130 PRINT"Magical Attack Adj..."
;MA
5140 PRINT"Open Doors on (d6)... "
;OD$

5150 CURSOR0,12:PRINT"You have a total
 of";GD;" Gold Pieces
5160 PRINT"to your name with which to
purchase  extra items & provisions.
5170 PRINT:PRINT"Your armour consists
of:-
5180 PRINT:PRINTTAB(4);AC$
5190 CURSOR5,22:PRINTPR$
5200 IN$=INKEY$:IFIN$=""THEN5200
5210 RETURN
5220 SCREEN1,1:CLS
5230 COLOR1,13
5240 PRINTTAB(8);"YOUR ABILITY SCORES
ARE:-
5250 PRINT TAB(7);"
5260 PRINT:PRINT"   STRENGTH....."
;ST;"/";SS:BEEP
5270 PRINT:PRINT"   INTELLIGENCE...."
;IN;" *":BEEP
5280 PRINT:PRINT"   WISDOM....."
;WI:BEEP
5290 PRINT:PRINT"   DEXTERITY....."
;DX:BEEP
5300 PRINT:PRINT"   CONSTITUTION...."
;CO:BEEP
5310 PRINT:PRINT"   CHARISMA....."
;CH:BEEP
5320 PRINT:PRINTTAB(3);I$
5330 PRINT:PRINT"Please copy these ont
o your character sheet, as they will n
ot appear again.
5340 PRINT"Once you have done that:- "
:PRINT:PRINTTAB(5);PR$
5350 IN$=INKEY$:IFIN$=""THEN5350
5360 CLS
5370 RETURN
5380 CLS
5390 PRINTEA$
5400 C=C+1
5410 W=INT(RND(-1)*40)+1
5420 X=5:Y=5
5430 CURSORX,Y+C:PRINTWE$(W):BEEP
5440 IFC=WETHEN5460
5450 GOTO5400
5460 CURSOR5,22:PRINTPR$
5470 IN$=INKEY$:IF IN$=""THEN5470
5480 C=0
5490 CLS
5500 PRINT"............MISC POSSESSIONS
............
5510 C=C+1
5520 P=INT(RND(-1)*75)+1
5530 X=5:Y=5
5540 CURSORX,Y+C:PRINTPO$(P):BEEP
5550 IFC=10THEN5570
5560 GOTO5510
5570 CURSOR5,22:PRINTPR$
5580 IN$=INKEY$:IF IN$=""THEN5580
5590 C=0
5600 GOTO6690
5610 CLS:PRINTTAB(7);CN$;" :-"
5620 PRINT:PRINTSK$
5630 PRINT:PRINTLL$
5640 C=C+1
5650 FA=INT(RND(-1)*12)+1
5660 X=0:Y=7
5670 CURSORX,Y+C:PRINTFA$(FA):BEEP
5680 IFC=L1THEN5700
5690 GOTO5640
5700 C=0:IFLL<3THEN5840
5710 C=C+1
5720 FB=INT(RND(-1)*12)+1
5730 X=13:Y=7
5740 CURSORX,Y+C:PRINTFB$(FB):BEEP
5750 IFC=L2THEN5770
5760 GOTO5710
5770 C=0:IFLL<5THEN5840
5780 C=C+1
5790 FC=INT(RND(-1)*11)+1
5800 X=25:Y=7
5810 CURSORX,Y+C:PRINTFC$(FC):BEEP
5820 IFC=2THEN5840
5830 GOTO5780
5840 CURSOR5,22:PRINTPR$
5850 IN$=INKEY$:IFIN$=""THEN5850
5860 C=0
5870 RETURN
5880 CLS:PRINTTAB(7);CN$;" :-"
5890 PRINT:PRINTSK$
5900 PRINT:PRINTLL$
5910 C=C+1
5920 FD=INT(RND(-1)*10)+1
5930 X=0:Y=7
5940 CURSORX,Y+C:PRINTFD$(FD):BEEP
5950 IFC=L1THEN5970
5960 GOTO5910
5970 C=0:IFLL=1THEN6110
5980 C=C+1
5990 FE=INT(RND(-1)*12)+1
6000 X=13:Y=7
6010 CURSORX,Y+C:PRINTFE$(FE):BEEP
6020 IFC=L2THEN6040
6030 GOTO5980
6040 C=0:IFLL=2THEN6110
6050 C=C+1
6060 FF=INT(RND(-1)*12)+1
6070 X=25:Y=7
6080 CURSORX,Y+C:PRINTFF$(FF):BEEP
6090 IFC=L3THEN6110
6100 GOTO6050
6110 CURSOR5,22:PRINTPR$
```

```
6120 IN$=INKEY$:IFIN$=""THEN6120
6130 C=0
6140 RETURN
6150 CLS:PRINTTAB(7);CN$;" :-"
6160 PRINT:PRINTSK$
6170 PRINT:PRINTLL$
6180 C=C+1
6190 FG=INT(RND(-1)*20)+1
6200 X=0:Y=7
6210 CURSORX,Y+C:PRINTFG$(FG):BEEP
6220 IFC=L1THEN6240
6230 GOTO6180
6240 C=0:IFLL=1THEN6380
6250 C=C+1
6260 FH=INT(RND(-1)*20)+1
6270 X=13:Y=7
6280 CURSORX,Y+C:PRINTFH$(FH):BEEP
6290 IFC=L2THEN6310
6300 GOTO6250
6310 C=0:IFLL=2THEN6380
6320 C=C+1
6330 FI=INT(RND(-1)*20)+1
6340 X=25:Y=7
6350 CURSORX,Y+C:PRINTFI$(FI):BEEP
6360 IFC=L3THEN6380
6370 GOTO6320
6380 CURSOR5,22:PRINTPR$
6390 IN$=INKEY$:IFIN$=""THEN6390
6400 C=0
6410 RETURN
6420 CLS:PRINTTAB(7);CN$;" :-
6430 PRINT:PRINTSK$
6440 PRINT:PRINTLL$
6450 C=C+1
6460 FJ=INT(RND(-1)*11)+1
6470 X=0:Y=7
6480 CURSORX,Y+C:PRINTFJ$(FJ):BEEP
6490 IFC=L1THEN6510
6500 GOTO6450
6510 C=0:IFLL<3THEN6650
6520 C=C+1
6530 FK=INT(RND(-1)*12)+1
6540 X=13:Y=7
6550 CURSORX,Y+C:PRINTFK$(FK):BEEP
6560 IFC=L2THEN6580
6570 GOTO6520
6580 C=0:IFLL<5THEN6650
6590 C=C+1
6600 FL=INT(RND(-1)*10)+1
6610 X=25:Y=7
6620 CURSORX,Y+C:PRINTFL$(FL):BEEP
6630 IFC=2THEN6650
6640 GOTO6590
6650 CURSOR5,22:PRINTPR$
6660 IN$=INKEY$:IFIN$=""THEN6660
6670 C=0
6680 RETURN
6690 CLS:COLOR15,1:CURSOR3,7:PRINT"I H
OPE YOU LIKE YOUR CHARACTER-":PRINT:PR
INT"Do you wish to generate another? (
Y/N)
6700 IN$=INKEY$:IFIN$=""THEN6700
6710 IFIN$="Y"THEN730
6720 IFIN$="N"THEN6730
6730 CURSOR3,15:PRINT"Enjoy your game
- Bye for now!!"
6740 PRINT:PRINT:PRINT:END
6750 DATA BATTLE AXE,THROWING AXE,SWOR
D SHORT,CLUB,DAGGER,BLOW PIPE,HAMMER,M
ORNING STAR,SWORD BROAD
6760 DATA CROSSBOW LIGHT,SWORD LONG,SC
IMITAR,SPEAR,SWORD TWO-HANDED
6770 DATA SHORT BOW,SWORD SHORT,DAGGER
,PICK,MACE,SCIMITAR,BO STICK,FAUCHARD,
MORNING STAR,SPEAR,TRIDENT,RANSEUR,SCI
MITAR,SWORD LONG,SWORD SHORT,SWORD BRO
AD
6780 DATA BATTLE AXE,SWORD BASTARD,VOU
LGE,SLING STONE,SLING BULLET,CROSSBOW
LIGHT,HALBERD,BARDICHE,DAGGER,DART
6790 DATA BELT,LEATHER BACKPACK,CANDLE
,PINT OF ALE,HOLY SYMBOL,ROPE 50ft,SAC
K SMALL,BELT POUCH,TORCH
6800 DATA PACK OF TOOTH PICKS,MIRROR S
MALL,POLE 10ft,SACK,ROPE 20ft,CANDLE,L
ANTERN,BACKPACK,PINT OF ALE,SACK
6810 DATA LARGE WOODEN CHEST,FLASK OF
OIL,PINT BEER,ROPE 30ft,POLE 10ft,HANK
ERCHIEF,CLOVE OF GARLIC,PINT WINE
6820 DATA BELT POUCH,BELT,SACK,SACK,LA
NTERN,HOLY SYMBOL,FLASK OF OIL,WATER S
KIN,IRON SPIKES,TORCH,SMALL CART
6830 DATA HAT,ROBE,STONE PICK,1 WEEKS
RATIONS,ROPE 40ft,TORCH,BELT,TORCH,SAC
K,WATER SKIN,IRON SPIKES
6840 DATA CAP,ROBE,CLOAK,BELT,BOOTS HI
GH,GARLIC BUD,WOLVESBANE,GIRDLE BROAD,
WAGON,PINT MEAD,BELT,CANDLE WAX,FLASK
OF OIL,TINDER BOX,SACK LARGE
6850 DATA PINT OF ALE,SMALL WOODEN CHE
ST,LEATHER BACKPACK,1 WEEKS RATIONS,QU
IVER 12 ARROWS,QUIVER 10 BOLTS,QUIVER
20 BOLTS,LANTERN BULLSEYE,BELT,PINT OF
 BEER
6860 DATABless,Command,Create Water,Cu
re L/Wound,Detect Evil,Detect Magic,Li
ght,Remove Fear,Resist Cold,Evil Prote
ct,Purify Food,Sanctuary
6870 DATAChant,Det Charm,Find Traps,Ho
ld Person,Resist Fire,Slow Poison,Snak
e Charm,Augury,Silence,Talk to Anim,Kn
ow Align.,Hammer
6880 DATAAnimate Dead,Cont Light,Praye
r,Dispel Magic,Feign Death,Cure Diseas
e,Find Object,Talk to Dead,Rem Curse,C
ure Blind,Create Food
6890 DATATalk to Anim,Purify Water,Pas
s,Entangle,Faerie Fire,Find Animals,De
tect Magic,Detect Pits,Invisible,Shill
elagh
```

```
6900 DATABarkskin,Make Water,Feign Dea
th,Fire Trap,Heat Metal,Find Plant,Mak
e Flame,Trip,Warp Wood,Heal Wounds,Cha
rm Person,Obscurement
6910 DATATree,Stone Shape,Snare,Pyrote
chnics,Plant Growth,Hold Animal,Cure D
isease,Call Insects,Fire Protect,Neut
Poison,Gills,Lightning
6920 DATA Push,Shield,Mending,Light,En
large,Hold Portal,Erase,Sleep,Jump,Jen
sers Disc,Charm Person,Feather Fall,Fr
iends,Write,Spider Climb,Read Magic
6930 DATAMessage,Evil Protect,Burn Han
ds,Identify
6940 DATA Web,Strength,Levitate,ESP,Sc
are,Fools Gold,Find Object,Shatter,Mir
ror Image,Magic Mouth,Light,Darkness,F
orget,Knock,Rope Trick,Wizard Lock
6950 DATAInvisible,Stink Cloud,Pyrotec
hnic,Leos Trap
6960 DATA Feign Death,Fly,Haste,Fireba
ll,Dispel Magic,Flame Arrow,Slow,Gust
Wind,Hold Person,Infravision,Suggestio
n,Tongues,Gills,Leos Hut,Invisible
6970 DATABlink,Clairvoyance,Phant Forc
e,Evil Protect,Lightning
6980 DATAAud Glamer,Change,Color Spray
,Lights,Darkness,Light,Fog,Reflection,
Hypnotism,Phant Force,Det Illusion
6990 DATABlindness,Blurr,Deafness,Fog
Cloud,Invisible,Mirror Image,Ventril.,
Magic Mouth,Misdirect,Det Magic,Hyp Pa
ttern,Phant Force
7000 DATARope Trick,Suggest,Spectral,P
aralize,Non-detect,Invisible,Fear,Cont
 Light,Cont Dark,Terrain


 559
Bytes free
```

HELICOPTER

By Jan Jacobson (C) 1986

```
10 REM ********************************
20 REM ***                          ***
30 REM ***       HELICOPTER         ***
40 REM ***                          ***
50 REM ***BY JAN JACOBSEN (C) 1986***
60 REM ***                          ***
70 REM ********************************
80 SCREEN 2,2:COLOR1,15,,15:CLS
90 RESTORE
100 REM LINE(57,56)-(207,132),1,B
110 REM COLOR 1,2 ,(60,56)-(200,132),15
120 R=R+1
130 READ X,Y
140 IF X=0 THEN READ X,Y:LINE(X,Y)-(X,Y)
150 IF X=-1 THEN 260
160 LINE-(X,Y)
170 GOTO 120
180 DATA 0,0,96,126,104,124,140,109,152,100,0,0,156,96,164,90,174,86,0,0,180,84,
188,81,192,76,0,0,185,73,192,60,184,62,176,76,156,84,152,85,148,85,138,85,0,0
190 DATA 114,90,108,92,104,94,100,96,96,99,92,104,89,109,88,111,84,113,81,116,80
,120,84,124,92,126,96,126,0,0
200 DATA 112,94,132,92,0,0,138,89,140,87,136,83,128,85,0,0,124,85,118,86,123,84,
0,0,182,89,186,87,176,82,172,85,182,89,0,0,163,82,160,80,164,78,168,80,0,0
210 DATA 191,68,182,84,184,85,193,68,191,68,0,0,104,94,112,100,114,118,0,0,113,1
05,140,94,140,108,0,0,123,94,127,97,127,100,0,0
220 DATA 90,125,127,113,127,115,128,100,0,0,116,108,116,114,126,110,126,104,116,
108,0,0,184,75,184,76,186,78,186,74,184,73,0,0,112,94,118,86,0,0
230 DATA 94,102,95,105,89,110,89,108,0,0,96,106,100,108,104,108,100,117,91,111,9
6,106,0,0,102,95,104,97,96,103,96,99,0,0,108,99,112,102,108,105,104,106,100,105,
108,99,0,0
240 DATA 152,87,154,89,156,92,156,96,0,0,144,91,154,89,0,0,144,87,148,87,152,86,
176,78,180,76,188,74,0,0,184,65,190,64,0,0
250 DATA 122,82,171,102,167,105,132,90,128,86,122,84,122,82,72,104,67,100,112,84
,122,82,80,66,82,62,118,78,127,82,176,62,180,65,136,82,122,82,-1,-1
260 PAINT(82,65),1:PAINT(70,101),1:PAINT(168,103),1:PAINT(176,64),1
270 PAINT(92,106),1:PAINT(96,108),1:PAINT(104,104),1:PAINT(100,98),1:PAINT(120,1
07),1:PAINT(184,68),6
280 GOTO 280
```

GOMOKU

By Jan Jacobsen 1986

```
10 REM GOMOKU BY JAN JACOBSEN
20 REM        1986
30 REM ADELAIDE SEGA USER CLUB
40 REM
45 REM
50 GOSUB720
60 GOSUB160
70 GOSUB290
80 S$=RIGHT$(STR$(G-1),2):X=(VAL(RIGHT$(S$,1))*16+14):Y=(VAL(LEFT$(S$,1))*16+14)
:CURSORX,Y:PRINT"k"
90 IFL>3THENCOLOR15:CURSOR192,64:PRINT"I WIN!!":FORA=1TO500:NEXT:END
100 GOTO 60
110 E=A
120 E=E+N:IFA(E)<>ZTHENRETURN
130 K=K+1:GOTO120
140 PRINT CHR$(11);
150 PRINT:PRINT
160 REM
170 CURSOR32,176:COLOR11:PRINT"Your move:":X=94:Y=176:G$=""
180 I$=INKEY$:IFI$=""THEN180
190 IFI$=CHR$(13)ANDLEN(G$)=2THENBEEP:BEEP:G=VAL(G$):GOTO240
200 IFI$=CHR$(8)ANDLEN(G$)>0THENGOSUB230:GOTO180
210 IFI$<"1"ORI$>"8"THEN180
220 BEEP:G$=G$+I$:CURSORX,Y:PRINTI$:X=X+8:GOTO180
230 X=X-8:BLINE(X,Y)-(X+7,Y+7),,BF:L=LEN(G$)-1:G$=LEFT$(G$,L):BEEP:RETURN
240 G=G+1
250 IFG<120RG>890RA(G)<>46THENBEEP2:GOTO170
260 Z=H
270 COLOR1:A(G)=Z:S$=RIGHT$(STR$(G-1),2):X=(VAL(RIGHT$(S$,1))*16+14):Y=(VAL(LEFT
$(S$,1))*16+14):CURSORX,Y:PRINT"I":BLINE(0,176)-(255,191),,BF
280 RETURN
290 A=G
300 L=0
310 FORX=1TO4:K=0:N=X(X)
320 GOSUB110
330 N=-N:GOSUB110
340 IFK>LTHENL=K
350 NEXT X
360 IFL>3THENCOLOR11:CURSOR192,64:PRINT"YOU WIN!!":FORA=1TO500:NEXT:END
370 T=1
380 IF T<>2THENZ=C
390 IF T=2THENZ=H
400 G=0:H1=0:L=0
410 FOR A=12TO89
420 M=0
430 IF A(A)<>46THEN540
440 FORX=1TO4:K=0:N=X(X)
450 GOSUB110
460 N=-N:GOSUB110
470 IFK>LTHENH1=0:L=K
480 IFL<>KTHEN510
490 IF T=1ANDL<4OR(T=2ORT=3)ANDL<2THEN510
500 M=M+1
510 NEXTX
520 IFM<=H1THEN540
530 H1=M:G=A
540 NEXT A
550 IFH1<>0THEN620
560 T=T+1:IFT<>4THEN380
```

```
570 A=1
580 G=INT(RND(1)*77)+13
590 IFA(G)=46THEN620
600 A=A+1:IFA<100THEN580
610 CURSOR32,176:COLOR15:PRINT "I CONCEDE THE GAME":PRINT"TO A MASTER!!":FORA=1T
0500:NEXT:END
620 A(G)=C
630 Z=C:A=G:L=0
640 FORX=1TO4
650 K=0
660 N=X(X)
670 GOSUB110
680 N=-N:GOSUB110
690 IFK>LTHENL=K
700 NEXTX
710 RETURN
720 CLS
730 DIMA(100),X(4)
740 FORC=1TO8
750 FORB=2TO9
760 A(C*10+B)=46
770 NEXTB
780 NEXTC
790 FORQ=1TO4
800 READZ:X(Q)=Z
810 NEXTQ
820 DATA1,9,10,11
830 H=ASC("I"):C=ASC("k")
840 SCREEN 2,2:CLS:PRINT CHR$(16):COLOR8:CURSOR95,10:PRINT CHR$(17);"GOMOKU":PRI
NTCHR$(16):COLOR1:CURSOR85,40:PRINT"BY JAN JACOBSEN":CURSOR120,50:PRINT"1986"
850 COLOR2:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT"     You'll find GOMOKU an
easy game to       learn, but one which is almost          impossible to win.
860 PRINT"     The computer plays extremly well in      this program.
870 COLOR1:CURSOR50,170:PRINT    "ENTER Y IF YOU WANT THE"
880 CURSOR50,180:PRINT "FIRST MOVE, N IF YOU DON'T"
890 N=0
900 N=N+1
910 A$=INKEY$
920 IFA$<>"y"ANDA$<>"Y"ANDA$<>"n"ANDA$<>"N"THEN900
930 DU=RND(-1)
940 SCREEN 2,2:COLOR1,8,,8:CLS
950 PRINTCHR$(17):PRINT:COLOR15
960 CURSOR200,10:PRINT"G"
970 CURSOR200,18:PRINT"O"
980 CURSOR200,26:PRINT"M"
990 CURSOR200,34:PRINT"O"
1000 CURSOR200,42:PRINT"K"
1010 CURSOR200,50:PRINT"U"
1020 PRINTCHR$(16)
1030 FORX=24TO19*8STEP16:LINE(24,X)-(19*8,X),1:LINE(X,24)-(X,19*8),1:NEXT:COLOR,
15,(24,24)-(19*8-1,19*8)
1040 S=1:FORA=27TO139STEP16:CURSOR10,A:PRINTS:CURSORA,16:PRINTS:S=S+1:NEXT
1050 IF A$="y"ORA$="Y"THENRETURN
1060 RESTORE:FORJ=1TOINT(RND(1)*12)+1
1070 READZ
1080 NEXTJ
1090 A(Z)=C
1100 S$=RIGHT$(STR$(Z),2):X=(VAL(RIGHT$(S$,1))*16+14):Y=(VAL(LEFT$(S$,1))*16+14)
:CURSORX,Y:PRINT"k"
1110 RETURN
1120 DATA34,35,44,46,47,54,55,56,57,66
```

CHESS

By Jan Jacobsen 1986

```
1 REM              CHESS
2 REM               1986
3 REM
4 REM                BY
5 REM
6 REM         JAN   JACOBSEN
7 REM
8 REM  ********************************
9 COLOR15,4
20 GOSUB 2970
30 GOTO 60
40 GOSUB 2580
50 GOSUB 2820
60 GOSUB 2580
70 REM ********************************
80 IF A$="S" THEN END
90 IF A$="X" THEN PRINT "EXCHANGING SIDES":GOSUB 3540:A$=""
100 IF A$="P" THEN GOSUB 3730
110 REM ********************************
120 FOR Z=1 TO 16:T(Z)=0:NEXT Z
130 U=0
140 PRINT "Please stand by..."
141 FOR W=1 TO 5:SOUND1,165,15:NEXT W:SOUND0:FOR W=1 TO 5:SOUND2,165,15:NEXT W:S
OUND0:FOR W=1 TO 5:SOUND3,165,15:NEXT W:SOUND0:FOR W=1 TO 30:SOUND1,139,15:NEXT
W:SOUND0
150 FOR Q=1 TO 64:IF A(S(Q))>=BB AND A(S(Q))<=RB THEN U=U+1:T(U)=S(Q):IF A(S(Q))
=KB THEN KM=S(Q)
160 NEXT Q:IF U<3 THEN GOTO 2230
170 GOTO 650
180 FOR Q=1 TO U:IF A(T(Q))=KB THEN T(Q)=T(U):T(U)=KM
190 NEXT Q
200 Q=INT(RND(1)*3)
210 IF A$="C" THEN Q=0
220 IF Q<U THEN Q=Q+1
230 Z=T(Q):GOSUB 280
240 IF MM=1 THEN GOSUB 2500:GOTO 40
250 IF Q<U THEN 220
260 GOTO 2360
270 REM ********************************
280 IFA(Z)=QBTHENGOSUB 910
290 IFA(Z)=RBTHENGOSUB 1170
300 IFA(Z)=BBTHENGOSUB 1420
310 IFA(Z)=NBTHENGOSUB 1690
320 IFA(Z)=PBTHENGOSUB 2240
330 RETURN
340 REM ********************************
350 IF A(X)=107 THEN MM=0:Q=Q+1:RETURN
360 IF X+9>88 THEN 380
370 IF A(X+9)<83 AND A(X+9)>65 AND RND(1)<.96 THEN RETURN
380 IF X-11<11 THEN 400
390 IF A(X-11)<83 AND A(X-11)>65 AND RND(1)<.96  THEN RETURN
400 AD=0
410 AY=0
420 AX=X+Q(AY+AD)
430 IF AX<11 OR AX>88 THEN 460
440 AP=A(AX)
450 IF AP=Q ORAP=R AND RND(1)>.8 OR AP=B AND RND(1)>.5 THEN RETURN
460 AY=AY+1
470 IF AY<8 THEN 420
```

```
480 AD=AD+7
490 IF AD<56 THEN 410
500 AY=1
510 AX=X+N(AY)
520 IF AX<11 OR AX>88 THEN 540
530 IF A(AX)=N THEN RETURN
540 AY=AY+1
550 IF AY<9 THEN 510
560 AY=1
570 AX=X +K(AY)
580 IF AX<11 OR AX>88 THEN 600
590 IF A(AX)=K OR A(AX)=P ANDRND(1)>.1 THEN RETURN
600 AY=AY+1
610 IF AY<9 THEN 570
620 MM=1
630 RETURN
640 REM ********************************
650 Z=KM
660 Y=0
670 Y=Y+1
680 X=Z+N(Y)
690 IF X<11 OR X>88 THEN 600
700 IF A(X)=N THEN 1870
710 IF Y<8 THEN 670
720 REM ********************************
730 D=0
740 Y=1
750 X=Z+Q(Y+D)
760 IF X<11 OR X>88 THEN 810
770 IF A(X)=B OR A(X)=Q OR A(X)=R THEN 1870
780 IF A(X)<>E THEN 810
790 Y=Y+1
800 IF Y<8 THEN 750
810 D=D+7
820 IF D<49 THEN 750
830 X=Z+11
840 IF X>88 THEN 860
850 IF A(X)=P THEN 1870
860 X=Z-11
870 IF X<11 THEN 180
880 IF A(X)=P THEN 1870
890 GOTO 180
900 REM ********************************
910 D=0
920 Y=1
930 X=Z+Q(Y+D)
940 IF X<11 OR X>88 THEN 1000
950 IF A(X)=42 OR A(X)>=BB AND A(X)<=RB THEN 1000
960 IF A(X)>=B AND A(X)<=R THEN GOSUB 350:IF MM<> 1 THEN 1000
970 IF MM=1 THEN RETURN
980 Y=Y+1
990 IF Y<7 THEN 930
1000 D=D+7
1010 IF D<42  THEN 920
1020 RETURN
1030 REM ********************************
1040 D=0
1050 Y=1
1060 X=Z+Q(Y+D)
1070 IF X<11 OR X>88  THEN 1130
1080 IF A(X)<>E THEN 1130
1090 IF RND(1)>.5 THEN GOSUB 350:IF MM=0 THEN GOTO 1130
1100 IF MM=1 THEN RETURN
1110 Y=Y+1
```

```
1120 IF Y<8 THEN GOTO 1060
1130 D=D+7
1140 IF D<49 THEN 1050
1150 RETURN
1160 REM ********************************
1170 D=0
1180 Y=1
1190 X=Z+R(Y+D)
1200 IF X<11 OR X>88 THEN 1260
1210 IF A(X)=42 OR A(X)>=BB AND A(X)<=RB THEN 1690
1220 IF A(X)>=B AND A(X)<=R THEN GOSUB 350:IF MM=0 THEN GOTO 1260
1230 IF MM=1 THEN RETURN
1240 Y=Y+1
1250 IF Y<7 THEN 1190
1260 D=D+7
1270 IF D<21 THEN 1180
1280 RETURN
1290 REM ********************************
1300 D=0
1310 Y=1
1320 X=Z+R(Y+D)
1330 IF X<11 OR X>88 THEN 1390
1340 IF A(X)<>E THEN 1390
1350 IF RND(1)<.1 THEN GOSUB 350
1360 IF MM=1 THEN RETURN
1370 Y=Y+1
1380 IF Y<7 THEN 1320
1390 D=D+7
1400 IF D<21 THEN 1310
1410 RETURN
1420 IF A(Z)<>BB  THEN RETURN
1430 D=0
1440 Y=1
1450 X=Z+B(Y+D)
1460 IF X<11 OR X>88 THEN 1520
1470 IF A(X)=42 OR A(X)>=BB AND A(X)<=RB THEN 1520
1480 IF A(X)>=B AND A(X)<=R  THEN GOSUB 350:IF MM<>1 THEN 1520
1490 IF MM=1 THEN RETURN
1500 Y=Y+1
1510 IF Y<7 THEN 1450
1520 D=D+7
1530 IF D<21 THEN 1440
1540 RETURN
1550 REM ********************************
1560 D=0
1570 Y=1
1580 X=Z+B(Y+D)
1590 IF X<11 OR X>88 THEN 1650
1600 IF A(X)<> E THEN 1650
1610 IF RND(1)>.05 THEN GOSUB 350:IF MM<>1 THEN 1650
1620 IF MM=1 THEN RETURN
1630 Y=Y+1
1640 IF Y<7 THEN 1580
1650 D=D+7
1660 IF D<21  THEN 1570
1670 RETURN
1680 REM ********************************
1690 Y=1
1700 X=Z+N(Y)
1710 IF X<11 OR X>88 THEN 1750
1720 IF A(X)=42 THEN 1750
1730 IF A(X)>=B AND A(X)<=R THEN GOSUB 350
1740 IF MM=1 THEN RETURN
1750 Y=Y+1
```

```
1760 IF Y<9 THEN 1700
1770 RETURN
1780 REM ********************************
1790 Y=0
1800 X=Z+N(INT(RND(1)*8)+1)
1810 IF X<11 OR X>88 THEN 1800
1820 IF A(X)=42 THEN 1800
1830 Y=Y+1
1840 IF A(X)=E THEN GOSUB 350
1850 IF MM=1 OR Y>20 THEN RETURN
1860 GOTO 1800
1870 YK=1
1880 Z=KM
1890 X=Z+K(YK):X1=X
1900 IF X<11 OR X>88 THEN 2200
1910 IF A(X)=42 OR A(X)>65 AND A(X)<83  THEN 2200
1920 IF A(X)>97 AND A(X)<115  THEN 2200
1930 Z=X
1940 Y=0
1950 Y=Y+1
1960 X=Z+N(Y)
1970 IF X<11 OR X>88 THEN 1990
1980 IF A(X)=N THEN 2200
1990 IF Y<8 THEN 1950
2000 REM ********************************
2010 D=0
2020 Y=1
2030 X=Z+Q(Y+D)
2040 IF X<11 OR X>88 THEN 2090
2050 IF A(X)=B OR A(X)=Q OR A(X)=R THEN 2200
2060 IF A(X)<>E  THEN 2090
2070 Y=Y+1
2080 IF Y<8 THEN 2030
2090 D=D+7
2100 IF D<49 THEN 2030
2110 X=Z+11
2120 IF X>88 THEN 2140
2130 IF A(X)=P THEN 2200
2140 X=Z-11
2150 IF X<11 THEN 2170
2160 IF A(X)=P THEN 2200
2170 X=X1:Z=KM
2180 MM=1
2190 GOSUB 2500:GOTO 40
2200 YK=YK+1
2210 Z=KM
2220 IF YK<9 THEN 1880
2230 PRINT "I concede, champ!":END
2240 X=Z+9
2250 IF A(X)>=B AND A(X)<=R THEN MM=1:IF A(X)=P AND RND(1)<.2 THEN MM=0
2260 IF MM=1 THEN RETURN
2270 IF Z=12 THEN RETURN
2280 X=Z-11
2290 IF A(X)>=B AND A(X)<=R THEN MM=1:IF A(X)=P AND RND(1)<.2 THEN MM=0
2300 RETURN
2310 REM ********************************
2320 IF Z-10*(INT(Z/10))=7 AND A(Z-1)=E AND A(Z-2)=E AND (A(Z-13)=E OR A(Z-13)=4
2) AND (A(+7)=E OR A(+7)=42) THEN X=Z-2:MM=1:RETURN
2330 IF A(Z-1)=E AND A(Z-12)<98 AND A(Z+8)<98 THEN X=Z-1:MM=1:RETURN
2340 IF RND(1)<.05 AND A(Z-1)=E THEN X=Z-1:MM=1
2350 RETURN
2360 Q=INT(RND(1)*RND(1)*5):IF Q>U THEN 2360
2370 IF Q<U THEN Q=Q+1
2380 Z=T(Q)
```

```
2390 IF A(Z)=PB THEN GOSUB 2320
2400 IF A(Z)=NB THEN GOSUB 1790
2410 IF A(Z)=BB THEN GOSUB 1560
2420 IF A(Z)=RB THEN GOSUB 1300
2430 IF A(Z)=QB THEN GOSUB 1040
2440 IF A(Z)=KB AND A$<>"C" AND RND(1)<.07 THEN GOSUB 1870
2450 IF MM=0 AND Q<U THEN GOTO 2370
2460 IF MM=1 THEN GOSUB 2500:GOTO 40
2470 UK=UK+1:IF UK>8 THEN 2230
2480 GOTO 2360
2490 REM ******************************
2500 IF A(Z)=KB AND A$<>"C" AND RND(1)>.1 THEN MM=0:GOTO 2360
2510 IF A(Z)=PB AND ((X-10*INT(X/10)>Z-10*INT(Z/10)OR ABS(X-Z)>11)) THEN MM=0:U=
U+1:GOTO 230
2520 IF A(X)=K THEN PRINT "CHECK":MM=0:U=U+1:GOTO 230
2530 A(X)=A(Z):A(Z)=E
2540 PRINT :PRINT "I will move from ";
2550 FZ=INT(Z/10):PRINT CHR$(FZ+64);Z-10*FZ;" to ";:FX=INT(X/10):PRINT CHR$( FX+
64);X-10*FX:FOR O=1 TO 1000:NEXT O
2560 RETURN
2570 REM ******************************
2580 CLS
2590 GOSUB 2670
2600 FOR X=8 TO 1 STEP -1
2610 PRINT TAB(10);X;" ";
2620 FOR Y=10 TO 80  STEP 10
2630 IF A(Y+1)=PB THEN A(Y+1)=QB
2640 IF A(Y+8)=P THEN A(Y+8)=Q
2650 PRINT CHR$(A(Y+X));" ";
2660 NEXT Y:PRINT X:NEXT X:MM=0
2670 PRINT:PRINT TAB(13);"A B C D E F G H":PRINT
2680 REM ******************************
2690 RETURN
2700 Z=KM
2710 QK=0
2720 M=Z+K(QK)
2730 IF M<11 OR M>88 THEN 2780
2740 IF A(M)=42 OR A(M)>65 AND A(M)<83 OR MM=0 THEN 2780
2750 X=M
2760 KM=Z
2770 RETURN
2780 IF QK<8 THEN 2720
2790 IF A$<>"C" THEN RETURN
2800 GOTO 2230
2810 REM ******************************
2820 PRINT
2830 INPUT "FROM (LETTER,NUMBER)- ";A$
2840 IF LEN(A$)<>2 THEN 2820
2850 PRINT:INPUT "TO- ";B$
2860 IF LEN(B$)<>2 THEN 2850
2870 X=10*(ASC(A$)-64)+VAL(RIGHT$(A$,1))
2880 Y=10*(ASC(B$)-64)+VAL(RIGHT$(B$,1))
2890 PRINT : PRINT "Enter C - check"
2900 PRINT "      P - to print board"
2910 PRINT "      x - to exchange side"
2920 PRINT "      s - to stop game"
2930 INPUT "Or press RETURN to continue";A$
2940 IF A(Y)>=75 AND A(Y)<=82  THEN GOSUB 3660
2950 A(Y)=A(X):A(X)=46:RETURN
2960 REM ******************************
2970 CLS:CURSOR 5, 4:PRINT "ARE YOU READY FOR ME TO START?":CURSOR8, 6:PRINT "IF
 SO THEN PRESS RETURN":PRINT:PRINT:PRINT
2980 N=N+1:IF INKEY$="" THEN 2980
2990 CLS:CURSOR12,12:PRINT"Please stand by"
```

```
3000 FOR L=1 TO 10:SOUND1,131,15:NEXT L:SOUNDO:FOR L=1 TO 10:SOUND2,147,15:NEXT
L:SOUNDO:FOR L=1 TO 10:SOUND3,131,15:NEXT L:SOUNDO:FOR L=1 TO 30:SOUND1,156,15:N
EXT L:SOUNDO
3010 MM=0:A$=""
3020 DIM A(99),R(28),B(28),N(8),Q(56),K(8),Z(88),S(64),T(16)
3030 P=112:R=114:N=110:B=98:Q=113:K=107:E=46
3040 PB =80:RB=82:NB=78:BB=66:QB=81:KB=75
3050 FOR Z=1 TO 99:A(Z)=-99:NEXT Z
3060 REM ********************************
3070 FOR Z=1 TO 64:READ X:READ Y:A(X)=Y:NEXT Z
3080 DATA 18,82,28,78,38,66,48,81
3090 DATA 58,75,68,66,78,78,88,82
3100 DATA 17,80,27,80,37,80,47,80
3110 DATA 57,80,67,80,77,80,87,80
3120 DATA 16,46,26,46,36,46,46,46
3130 DATA 56,46,66,46,76,46,86,46
3140 DATA 15,46,25,46,35,46,45,46
3150 DATA 55,46,65,46,75,46,85,46
3160 DATA 14,46,24,46,34,46,44,46
3170 DATA 54,46,64,46,74,46,84,46
3180 DATA 13,46,23,46,33,46,43,46
3190 DATA 53,46,63,46,73,46,83,46
3200 DATA 12,112,22,112,32,112,42,112
3210 DATA 52,112,62,112,72,112,82,112
3220 DATA 11,114,21,110,31,98,41,113
3230 DATA 51,107,61,98,71,110,81,114
3240 REM ********************************
3250 RESTORE 3270
3260 FOR Z=1 TO 8:READ N(Z):NEXT Z
3270 DATA 19,-19,21,-21,-8,8,12,-12
3280 FOR Z=1 TO 28:READ R(Z):NEXT Z
3290 DATA 10,20,30,40,50,50,50
3300 DATA -1,-2,-3,-4,-5,-5,-5
3310 DATA -10,-20,-30,-40,-50,-50,-50
3320 DATA 1,2,3,4,5,5,5
3330 RESTORE 3350
3340 FOR Z=1 TO 28:READ B(Z):NEXT Z
3350 DATA -11,-22,-33,-44,-55,-55,-55
3360 DATA 11,22,33,44,55,55,55
3370 DATA 9,18,27,36,45,45,45
3380 DATA -9,-18,-27,-36,-45,-45,-45
3390 RESTORE 3290
3400 FOR Z=1  TO 56:READ Q(Z):NEXT Z
3410 FOR Z=1 TO 8:READ K(Z):NEXT Z
3420 DATA 1,11,9,10,-10,-9,-11,-1
3430 FOR Z=1 TO 64:READ S(Z):NEXT Z
3440 DATA 46,56,36,66,47,57,45,55
3450 DATA 37,67,35,65,28,78,27,77
3460 DATA 44,54,26,76,38,68,17,87
3470 DATA 18,88,34,64,25,75,16,86
3480 DATA 48,24,74,15,85,14,84,43
3490 DATA 53,33,62,23,73,52,42,62
3500 DATA 32,83,13,72,22,12,82,41
3510 DATA 51,31,61,21,71,11,81,58
3520 CLS:RETURN
3530 REM ********************************
3540 FOR Z=11 TO 88:Z(Z)=A(Z):NEXT Z
3550 FOR Z=11 TO 88:X=Z-10*INT(Z/10)
3560 IF X=0 OR X=9  THEN 3580
3570 A(Z)=Z(Z+9-X*2)
3580 NEXT Z
3590 FOR Z=11 TO 88:M=A(Z)
3600 IF M>=B THEN A(Z)=A(Z)+PB-P
3610 IF M<=RB AND M>=BB THEN A(Z)=A(Z)-PB+P
```

```
3620 NEXT Z
3630 GOSUB 2580
3640 RETURN
3650 REM ****************************
3660 CM=INT(RND(1)*4)+1
3670 ON CM GOSUB 3690,3700,3710,3720
3680 FOR J=1 TO 1000:NEXT J:RETURN
3690 PRINT : PRINT " Well done !":RETURN
3700 PRINT : PRINT " Good move ":RETURN
3710 PRINT:PRINT " Great move ,champ !":RETURN
3720 PRINT :PRINT " Got me...":RETURN
3730 LPRINT "****************************"
3740 GOSUB 3800
3750 FOR X=8 TO 1 STEP -1
3760 LPRINT TAB(5);X;" ";
3770 FOR Y=10 TO 80 STEP 10
3780 LPRINT CHR$(A(X+Y));" ";
3790 NEXT Y:LPRINT X:NEXT X
3800 LPRINT :LPRINT TAB(9);"A B C D E F G H":LPRINT
3810 RETURN
```

EGGBERT

By Mark Fisher

```
10 REM*************************************
20 REM*************************************
30 REM***                                ***
40 REM*** By Mark'EGGBERT' Fisher        ***
50 REM***                                ***
60 REM***  of the S.A. User Club         ***
70 REM***                                ***
80 REM*************************************
90 REM*************************************
100 REM
110 COLOR15,4
120 CLS
130 PRINTTAB(15);"EGGBERT"
140 PRINT"---------------------------------------"
150 INPUT"TRACK  ";T
160 IF T<OORT>39THENBEEP2:GOTO 150
170 INPUT"SECTOR ";S
180 IF S<1ORS>16THENBEEP2:GOTO 170
190 DSKI$ T,S;A$,0,128;B$,128,128
200 CLS:PRINT"   ";:FORX=0TO15
210 PRINTRIGHT$(HEX$(X),1);:NEXT:PRINT:PRINT
220 FORY=0TO20:CURSOR22,Y:PRINT"":NEXT:CURSOR0,2
230 FORY=0TO7:FORX=0TO15
240 IFX=0THENPRINTY
250 CURSORX+3,Y+2
260 I$=MID$(A$,X+Y*16+1,1)
270 IF I$>=" "THENPRINTI$:GOTO 290
280 PRINT"."
290 NEXT X,Y
300 FORY=0TO7:FORX=0TO15
310 IFX=0THENPRINT" ";HEX$(Y+8)
320 CURSORX+3,Y+8+2
330 I$=MID$(B$,X+Y*16+1,1)
340 IF I$>=" "THENPRINTI$:GOTO 360
350 PRINT"."
360 NEXT X,Y
370 PRINT:PRINT"   ";:FORX=0TO15
380 PRINTRIGHT$(HEX$(X),1);:NEXT
390 CURSOR25,2:PRINT"-A/  ASCII"
400 CURSOR25,3:PRINT"-D/  DEC"
```

```
410 CURSOR25,4:PRINT"-H/  HEX"
420 CURSOR25,5:PRINT"-Q/  QUIT"
430 CURSOR25,6:PRINT"-W/  WRT"
440 CURSOR25,7:PRINT"-S/  NEXT"
450 CURSOR25,8:PRINT"     SECTOR"
460 CURSOR25,9:PRINT"-M   HEX"
470 CURSOR25,10:PRINT"     DISPLAY"
480 CURSOR25,13:PRINT"TRACK  ";T
490 CURSOR25,14:PRINT"SECTOR ";S
500 CURSOR0,21:INPUT"START ";S$
510 S1=VAL("&H"+MID$(S$,2,1)+MID$(S$,1,1))
520 ED=0
530 IF RIGHT$(S$,2)="-A"THENED=1
540 IF RIGHT$(S$,2)="-D"THENED=2
550 IF RIGHT$(S$,2)="-H"THENED=3
560 IF RIGHT$(S$,2)="-Q"THEN120
570 IF RIGHT$(S$,2)="-W"THENGOSUB 790:GOTO 120
580 IF RIGHT$(S$,2)="-S"THENS=S+1:IFS>16THENT=T+1:IFT>39THENT=0:GOTO 190
590 IF RIGHT$(S$,2)="-S"THEN190
600 IF RIGHT$(S$,2)="-M"THENED=4
610 IF ED=0THENCURSOR0,22:INPUT"EDIT MODE ";M$:S$=S$+M$:GOTO 520
620 CURSOR0,21:PRINTSPC(38)
630 CURSOR0,21:INPUT"<GO> ";ED$
640 CURSOR0,21:PRINTCHR$(5)
650 CURSOR0,21:PRINT"PLEASE WAIT..."
660 ONEDGOSUB 690,720,760,840
670 GOTO 200
680 END
690 IFS1>127THENS1=S1-128:GOTO 710
700 EA$=LEFT$(A$,S1)+ED$+RIGHT$(A$,128-S1-LEN(ED$)):A$=EA$:RETURN
710 EA$=LEFT$(B$,S1)+ED$+RIGHT$(B$,128-S1-LEN(ED$)):B$=EA$:RETURN
720 REM
730 FORR=1TOLEN(ED$) STEP 3
740 V=VAL(MID$(ED$,R,3)):E$=E$+CHR$(V)
750 NEXT:ED$=E$:GOSUB 690:RETURN
760 FORR=1TOLEN(ED$) STEP 2
770 V=VAL("&H"+MID$(ED$,R,2)):E$=E$+CHR$(V)
780 NEXT:ED$=E$:GOSUB 690:RETURN
790 CLS
800 INPUT"ARE YOU SURE (Y/N) "; R$
810 IF R$<>"Y"THENRETURN
820 DSKO$ T,S;A$,0,128;B$,128,128
830 RETURN
840 CLS
850 FORB1=1TO128 STEP 8
860 FORB2=B1TOB1+7
870 PU$=HEX$(ASC(MID$(A$,B2,1)))+" "
880 IFLEN(PU$)<3THENPU$="0"+PU$
890 PRINTPU$;
900 NEXT
910 PRINT:NEXT
920 PRINT:PRINT"HIT ANY KEY TO CONTINUE"
930 IF INKEY$=""THEN930
940 CLS
950 FORB1=1TO128 STEP 8
960 FORB2=B1TOB1+7
970 PU$=HEX$(ASC(MID$(B$,B2,1)))+" "
980 IFLEN(PU$)<3THENPU$="0"+PU$
990 PRINTPU$;
1100 NEXT
1110 PRINT:NEXT
1120 PRINT:PRINT"HIT ANY KEY TO CONTINUE"
1130 IF INKEY$=""THEN1130
1140 RETURN
```

FILE SIZE CHECK PROGRAM

By Alex Farkas & Jan Jacobsen 1986
(For Disk Only)

```
10 REM File Size Check Program
20 REM            BY
30 REM
40 REM ALEX FARKAS & JAN JACOBSEN
50 REM
60 REM          1986
70 REM
80 REM     (for disk only)
90 REM
100 CONSOLE0,24:CLS:PRINT"ALJAPA file size check program Mk.IX  ----------------
----------------------":PRINT"File:  Name:";TAB(17);"Size:";TAB(25);"Type:":PRINT
"--------------------------------------"
110 ERASE:CONSOLE5,17:CLS
120 M$(1)="Push <SPACE> to continue"
130 M$(2)="Push <R> to RUN program."
140 M$(3)="  Push <K> to KILL file"
150 M$(4)="  Push <S> to SET file"
160 M$(5)="Push <N> to RENAME file"
170 M$(6)=" Push <L> to LOADM file"
180 DIMP$(192)
190 DATA F3,0E,14,06,0D,11,00,FF,CD,A7,01,CD,AD,01,CD,95,01,C9
200 RESTORE190:FORX=&HFE00TO&HFE11:READA$:POKEX,VAL("&H"+A$):NEXT:CALL&HFE00:GOS
UB 510
210 CF=0
220 N=1:FL=0:FORS=1TO12:DSKI$20,S;A1$,0,128;A2$,128,128
230 FORCH=0TO7:CH$=MID$(A1$;CH*16+1,12)
240 IFLEFT$(CH$,1)=CHR$(0)THENN=N-1:GOTO280
250 PRINTN;TAB(4);":";CH$;:CL$=MID$(A1$,(CH*16)+13,2):P$(N)=CH$
260 K=0:TY=ASC(LEFT$(CL$,1)):CL=ASC(RIGHT$(CL$,1)):H=TY:GOSUB410
270 IFCF+UC=&HA0THENCH=8:NEXTCH:S=20:GOTO370
280 N=N+1:NEXTCH
290 FORCH=0TO7:CH$=MID$(A2$,CH*16+1,12)
300 IFLEFT$(CH$,1)=CHR$(0)THENN=N-1:GOTO340
310 PRINTN;TAB(4);":";CH$;:CL$=MID$(A2$,(CH*16)+13,2):P$(N)=CH$
320 K=0:TY=ASC(LEFT$(CL$,1)):CL=ASC(RIGHT$(CL$,1)):H=TY:GOSUB410
330 IFCF+UC=&HA0THENCH=8:NEXTCH:S=20:GOTO370
340 N=N+1:NEXTCH
350 FL=0:GOSUB580:IFFL=1THENS=30:NEXTS:GOTO790
360 IFFL>1THENS=30:NEXTS:GOSUB790:GOTO110
370 NEXTS
380 FL=0:GOSUB580:IFFL=1THEN790
390 IFFL>1THENGOSUB790:GOTO110
400 CONSOLE4,20:CURSOR0,22:PRINT"Disk has";DSKF;" Kbytes free":END
410 CF=CF+1:K=K+1:A$=RIGHT$("0"+HEX$(PEEK(&HFF00+TY)),2):TY=VAL("&H"+A$)
420 IFLEFT$(A$,1)<>"C"THEN410
430 PRINTTAB(18);K;"k";TAB(25);
440 IFCL=0THENT$="  Basic"
450 IFCL=&H80THENT$="* Basic"
460 IFCL=2THENT$="  HEX file"
470 IFCL=&H82THENT$="* HEX file"
480 IFCL=1THENT$="  ASCII"
490 IFCL=&H81THENT$="* ASCII"
500 PRINTT$:RETURN
510 DATA F3,11,00,FF,06,A0,1A,13,FE,FF,28,07,FE,FE,28,0D,10,F4,C9,3A,FE,FE,3C,32
,FE,FE,10,EA,C9,3A,FD,FE,3C,32,FD,FE,10,E0,C9,*
520 X=&HFE80
530 READA$:IFA$<>"*"THENPOKEX,VAL("&H"+A$):X=X+1:GOTO530
540 POKE&HFEFE,0:POKE&HFEFD,0
550 CALL&HFE80
560 UC=PEEK(&HFEFE)+PEEK(&HFEFD)
570 RETURN
```

```
580 CONSOLE5,19:H=0
590 H=H+1:IFH=7THENH=1
600 CURSOR8,23:PRINTM$(H);
610 FORU=1TO30:IFINKEY$=" "THEN770
620 IFINKEY$="R"THENU=40:NEXTU:FL=1:GOTO790
630 IFINKEY$="K"THENU=40:NEXTU:FL=2:RETURN
640 IFINKEY$="S"THENU=40:NEXTU:FL=3:RETURN
650 IFINKEY$="N"THENU=40:NEXTU:FL=4:RETURN
660 IFINKEY$="L"THENU=40:NEXTU:FL=5:RETURN
670 NEXTU
680 CURSOR0,23:PRINTCHR$(21);
690 FORU=1TO30:IFINKEY$=" "THEN770
700 IFINKEY$="R"THENU=40:NEXTU:FL=1:GOTO790
710 IFINKEY$="K"THENU=40:NEXTU:FL=2:RETURN
720 IFINKEY$="S"THENU=40:NEXTU:FL=3:RETURN
730 IFINKEY$="N"THENU=40:NEXTU:FL=4:RETURN
740 IFINKEY$="L"THENU=40:NEXTU:FL=5:RETURN
750 NEXTU
760 GOTO 590
770 U=30:NEXTU:CLS:BEEP:RETURN
780 FL=1:RETURN
790 BEEP:CURSOR0,23:PRINT CHR$(21);:INPUT"Number please:";N
800 IFP$(N)=""THENBEEP2:GOTO790
810 POKE&HFE00,&H11:POKE&HFE01,&H20
820 POKE&HFE02,&HFE:POKE&HFE03,&HC3
830 ONFLGOTO840,870,890,920,940
840 POKE&HFE04,&HD0:POKE&HFE05,&H05
850 P$(N)=" """+P$(N)+""""+CHR$(13)+CHR$(0)
860 FORA=1TOLEN(P$(N)):POKE&HFE1F+A,ASC(MID$(P$(N),A,1)):NEXT:CONSOLE0,24:CLS:BE
EP:CALL&HFE00
870 CLS:KILLP$(N)
880 RETURN
890 INPUT"Condition:";C$:CLS
900 IFC$="P"THENSETP$(N),"P":RETURN
910 SETP$(N),"O":RETURN
920 PRINTP$(N):INPUT "New name:";NN$:CLS
930 NAME P$(N) AS NN$:RETURN
940 PRINT "LOADM :-";P$(N);" into?"
950 INPUT ">";S$:S=VAL("&H"+S$):LOADM P$(N),S:CLS:RETURN
```

MUSIC BIT

By Paul Schwarz 1986

```
10 REM
20 REM MUSIC BIT
30 REM            BY
40 REM
50 REM               PAUL
60 REM                    SCHWARZ
70 REM                          1986
80 REM
90 REM     ADELAIDE SEGA USER CLUB
100 REM
110 REM
120 REM
130 ERASE
140 DIM F0(10),F1(10),F2(10),R0(10),V(10),L(10),A(10),B(10)
150 CLS
160 PRINT"  DO YOU WANT TO "
170 PRINT"  ££££££££££££"
180 PRINT:PRINTTAB(08);"DEFINE FREQUENCY  (1)"
190 PRINT:PRINTTAB(08);"DEFINE NOISE      (2)"
200 PRINT:PRINTTAB(08);"QUIT THE PROGRAM  (3)"
```

```
210 PRINT:PRINT TAB (8);:INPUT "YOUR SELECTION  ";A
220 ON A  GOTO  230,600,890
230 CL=3840000
240 CLS:INPUT "HOW MANY FREQUENCIES (MAX=20) ? ";NF:IF NF<1 OR NF>10 THEN GOTO 2
40
250 PRINT:FOR R=1 TO NF:PRINT"ENTER FREQ.,VOICE,VOL.";R;" :";:INPUT F0(R),V(R),L
(R):NEXT
260 FOR R=1 TO NF
270 R0(R)=INT(CL/(32*F0(R)))
280 IF R0(R)<1 OR R0(R)>9999 THEN GOSUB900
290 F1(R)=CL/(32*R0(R))
300 F2(R)=CL/(32*(R0(R)+1))
310 IF ABS(F0(R)-F1(R))<ABS(F0(R)-F2(R)) THEN GOTO 330
320 AC(R)=F2(R):R0(R)=R0(R)+1:GOTO 340
330 AC(R)=F1(R)
340 B(R)=INT(R0(R)/16)
350 A(R)=R0(R)-B(R)*16+128
360 A(R)=A(R)+32*(V(R)-1)
370 VL=15-L(R)
380 VB(R)=144+VL+32*(V(R)-1)
390 NEXT
400 CLS:PRINT"FRQ.   REG  VCE  VCB1  VCB2  VOL  VCB"
410 PRINT"ggggggggggggggggggggggggggggggggggg":CONSOLE 2,22,,,2
420 FOR R=1 TO NF:PRINTTAB(0);F0(R);TAB(6); R0(R) ;TAB(12);V(R);TAB(17);"h";HEX$
(A(R));TAB(23);"h";HEX$(B(R));TAB(28);L(R);TAB(34);"h";HEX$(VB(R))
430 PRINT"£££££££££££££££££££££££££££££"
440 NEXT
450 PRINT"HARD COPY = 'H'      'CR'=CONTINUE"
460 A$=INKEY$:IF INKEY$=""THEN 460
470 IF A$="H" THEN HCOPY
480 CONSOLE 0,24:CLS
490 PRINT"WOULD YOU LIKE TO HEAR SOUND (Y/N)? "
500 INPUT A$:IF A$="" THEN GOTO500
510 IF A$="N" THEN GOTO 130
520 IFA$<>"Y"THEN 490
530 IF A$="N" THEN GOTO 150
540 PRINT:PRINT"WHAT TIME DELAY";:INPUT TD
550 FOR R=1 TO NF:OUT &H7F,A(R):OUT &H7F,B(R):OUT &H7F,VB(R):FOR W=1 TO TD:NEXT
W:NEXT R:SOUND0
560 PRINT"REPEAT (Y/N) ";:INPUT A$
570 IF A$="N" THEN GOTO 130
580 IF A$="Y" THEN GOTO 540
590 GOTO 560
600 CLS
610 PRINT"SELECT WHITE NOISE (1) OR PERIODIC (2)"
620 REM
630 INPUT FB
640 NB=224+FB*4
650 PRINTTAB(15);"SELECT HIGH PITCH (0)"
660 PRINTTAB(15);"  '' MEDIUM PITCH (1)"
670 PRINTTAB(15);"  ''    LOW PITCH (2)"
680 PRINT"PITCH UNDER CONTROL OF VOICE 3   (3)"
690 PRINT:PRINTTAB(18);:INPUT "PITCH SELECTION ";NF
700 IF NF<0 OR NF>3 THEN 620
710 NB=NB+NF
720 PRINT
730 PRINT"WHAT VOLUME DO YOU WANT (15 TO 0);
740 INPUT V
750 PRINT
760 V=15-V
770 NV=240+V
780 PRINT"NOISE CONTROL BYTE   (SENT FIRST)=&H";HEX$(NB)
790 PRINT"VOLUME CONTROL BYTE (SENT SECOND)=&H";HEX$(NV)
800 PRINT:PRINT:PRINT"ENTER 'H' TO HEAR SOUND"
```

```
810 PRINT:PRINT"ENTER 'C' TO OBTAIN HARD COPY"
820 INPUT"OR 'R' TO RETURN TO MENU";A$
830 IF A$="R"THEN GOTO130
840 IF A$="C" THEN HCOPY:GOTO 800
850 OUT &H7F,NB:OUT &H7F,NV:FOR W=1 TO 200:NEXT:SOUND 0
860 PRINT"REPEAT (Y/N)";:INPUT A$
870 IF A$="Y" THEN GOTO 850
880 GOTO 130
890 CLS:END
900 BEEP:BEEP:RETURN
```

cont from p21

```
4660 COLOR1:CURSOR203,2:PRINT LI
4670 BEEP:RETURN
4680 CLS:SCREEN 1,1:COLOR7,1:CLS
4690 PRINT" In the first screen, you are fightingoff dangerous choppers that won
't stopfiring at you and to make things even worse, lighting bolts can come out
of the sky."
4700 PRINT" The second screen is the same as the first but your jet is facing th
e otherway and you are fighting off jets thatare just as deadly as the choppers.
   The points are at random in these two rounds."
4710 PRINT" In the third round, you are a tank onthe surface, fighting off jets,
 jumbos, and choppers, are are all as deadly are each other.You get the score ta
blefor that round only, and the lighting bolts are still there"
4720 PRINT" You get 20 men in this game and at   the end of rounds 1 & 2, you ge
t bonuspoints. Use the Joy-stick to move up & down and left & right every third
   round. Be careful of sparks after exp-losion. Extra man at 5000"
4730 IF INKEY$="" THEN 4730
4740 RETURN
```

cont from p14

TABLE XX25 VIDEO PORT CONNECTIONS

Pin Number	*Function*
1	Audio
2	Gnd
3	Video
4	Gnd
5	Gnd

EXPANSION EDGE CONNECTOR

Pin Number		*Pin Number (Component side)*	
1	A0	1	+5v
2	A1	2	+5v
3	A2	3	CSRAM *
4	A3	4	CER0M2 *
5	A4	5	MEMRD *
6	A5	6	MEMWR *
7	A6	7	I/ORD *
8	A7	8	I/OWR *
9	A8	9	No Connection
10	A9	10	MREQ *
11	A10	11	CON
12	A11	12	RAS1 *
13	A12	13	CAS1 *
14	A13	14	RAM A7
15	D0	15	RAS2 *
16	D1	16	CAS2 *
17	D2	17	MUX *
18	D3	18	A14
19	D4	19	A15
20	D5	20	No Connection
21	D6	21	GND
22	D7	22	GND

NOTE: *means active low

CP/M 2.2 for SEGA

CP/M 2.2 will enable users to access CP/M programs on Amtrad CP/M 2.2 disks.

$AUS 52.00 ***(incl p&h)***

from Jan Jacobsen or Les Beacall — see Australian User Groups

USER GROUPS

AUCKLAND (NTH SHORE) SEGA USER GROUP
Contact: Norman Raynel
78 Anzac St
Takapuna
AUCKLAND
Ph: 496-841

PUKEKOHE SEGA USER GROUP
Contact: Selwyn Easton
4 Roose Ave
PUKEKOHE
Ph: 86-583

SOUTH COROMANDEL SEGA USER GROUP
Contact: Sid Hawken
PO Box 183
WHANGAMATA
Ph: 58-775

HAMILTON SEGA USER GROUP
PO Box 1548 Hamilton
President: Mr Colin Bell Ph: 73-826
Secretary: Mrs Anne Thrush Ph: 437-312
Meetings held 2nd & 4th Monday of each month at Whitiora Community Centre, Willoughby St at 7.30pm.

TOKOROA SEGA USER GROUP
Contact: Geoff Crawford
1 Pio Pio Pl
TOKOROA
Ph: 67-105

ROTORUA SEGA USER GROUP
Contact: Terry Cole
PO Box 7140
Te Ngae
ROTORUA
Ph: 59-325 AH

WHAKATANE SEGA USER GROUP
Contact: K Nightingale
240 King St
WHAKATANE
Ph: 84-500

NAPIER SEGA USER GROUP
Contact: Reid Duncan
Store Manager
Agnews Refrigeration
Ph: 55-857, 57-431

HAWERA SEGA USER GROUP
Contact: D.M. Beale
7A Clive St
HAWERA
Ph: 85-108

MARTON SEGA USER GROUP
Contact: Mr H.R. Miller
41 Alexandra St
MARTON

WELLINGTON SEGA USER GROUP
Contact: Shaun Parsons
PO Box 1871
WELLINGTON
Ph: 897-095 (AH) 727-666 (BUS)

CHRISTCHURCH SEGA USER GROUP
Contact: James O'Donnell
15 Jebson St
Shirley
CHRISTCHURCH
Ph: 856-884

TIMARU SEGA USER GROUP
Contact: John Oliver
South Canterbury Computer User Group
PO Box 73
TIMARU
Ph 26-300

OAMARU SEGA USER GROUP
Contact: Bill Dowman
99 Aln St
OAMARU
Ph: 46-250

DUNEDIN SEGA USER GROUP
Central City Computer Interest Group
Box 5260
Moray Place
DUNEDIN
Contact: Graeme Simpson,
Saddle Hill, R.D.1, Dunedin.
Ph: (089) 6374

HI-SCORE CHALLENGE

Borderline	16,802,820	Michael Wilkinson
Congo Bongo	175,490	David Downs
Flicky	2,718,900	B. A. Smaill
Lode Runner	38,000	Steve Biggs
Monaco GP	210,062	Stewart Parkes
Pacar	1,495,500	Tony Sasso
Pop Flamer	117,000	Richard Hendra
Sinbad Mystery	190,000	Stewart Parkes
Star Jacker	398,000	Steve Biggs
Video Flipper	999,880	Andre Stokes
Yamoto	85,000	Stewart Parkes
N-Sub	76,250	Reuban Jackson
Safari Hunting	8,400	David Downs
Safari Race	30,000	Jonathan Fletcher
Orguss	18,500	David Downs
Champion Tennis	9-Love	David Downs

Challenge these Hi-Scores for Cartridge games by sending us yours.

OVERSEAS SEGA USERS CLUBS

Klaus Pinker
P.O. Box 18
Belconnen
ACT 2616
Ph: 062 30 2334

Scott McDonald
2 Coolalie Ave
Camden
NSW 2570
Ph: 046 668 956

John McLennan
65 Highclere Blvd
Marangaroo
WA 6064
Ph: 09 342 3905

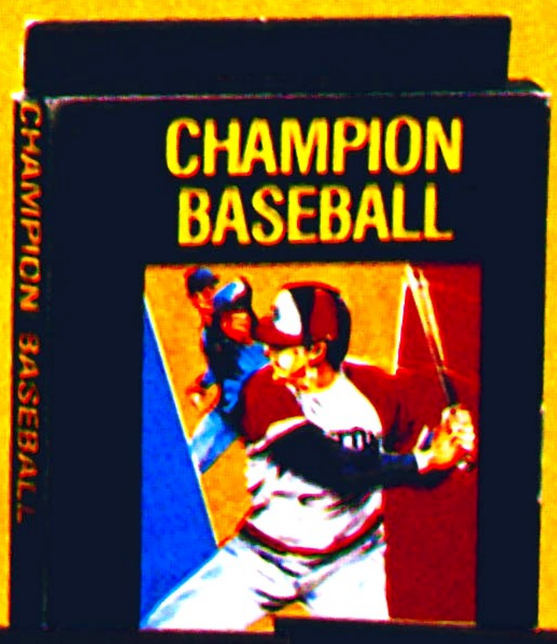
Jan Jacobson
10 Pioneer Ave
O'Sullivans Beach
SA 5166
Ph: 08 382 7967

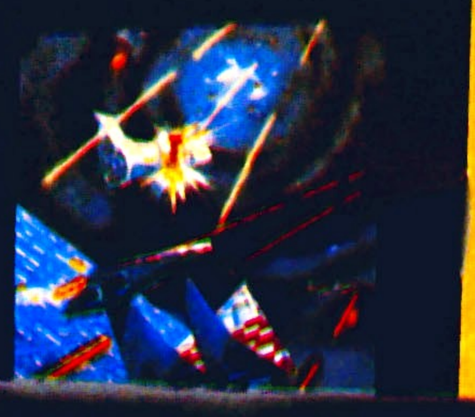
Les Beacall
1/41 Cameron Road
Croydon
VIC 3136
Ph: 03 725 0864

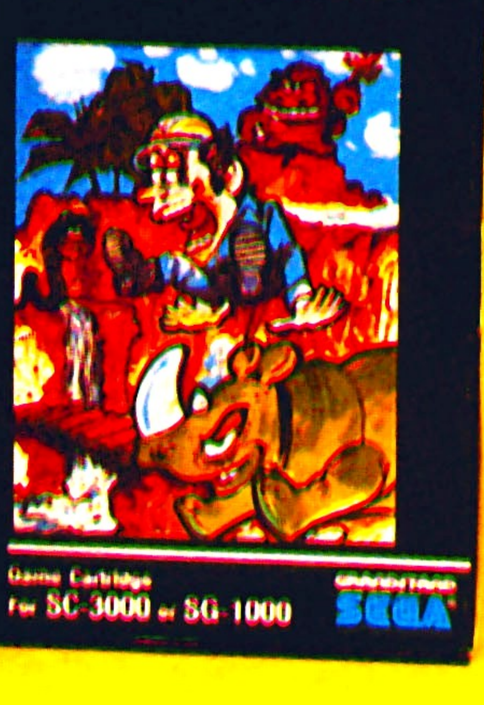
Wayne Ariel
35 Leanne Street
Marsden
QLD 4203

SEGA

GAMES CARTRIDGE SOFTWARE







701 BORDERLINE
Manoeuvre your jeep forward and backward to fight the enemy as you seek to destroy their fortresses and headquarters. Real action.

702 CHAMPION BASEBALL
Squeeze, steal, and double play.....!! play the computer or your friends. Very enjoyable.

703 CHAMPION TENNIS
Serve, receive and smash. Attack your rival's weak point. Very competitive.

704 CONGO BONGO
Watch out for falling coconuts and pesky monkeys as you attempt to preserve the hunter's honour and capture the gorilla.

705 MONACO GP
The race of your life with you at the wheel. Avoid the other cars, the ambulance etc and above all watch those sign posts!! Exciting

706 PACAR
Sega's own version of Pacman. Can you keep up the torrid pace while destroying as many enemy cars as possible?

707 POP FLAMER
A mouse that operates a flame thrower! 3 monsters to be eliminated here. Be especially wary of the Haradon.

708 SINBAD MYSTERY
Enjoy the cheerful adventures of Sinbad in his search for a chest of gold hidden on Treasure Island.

709 STAR JACKER
A "new feeling" space game. Glorious space troops fight against the invading "Dester" army. Fun for all the family.

710 VIDEO FLIPPER
Pinball fun for all. A colourful exciting game of skill needing split second timing. Even your five year old will love this.

711 YAMOTO
A fierce battle against the enemy's torpedoes and missiles. Can you survive - you'll need all your wits about you.

Only Available from SEGA SOFTWARE SUPPORT

P.O. Box 10-207
Auckland 4 New Zealand